JOURNAL OF APPLIED CRYSTALLOGRAPHY

ISSN: 1600-5767

journals.iucr.org/j

XDSAPP2.0

Karine M. Sparta, Michael Krug, Udo Heinemann, Uwe Mueller and Manfred S. Weiss

J. Appl. Cryst. (2016). **49**, 1085–1092

JOURNAL OF APPLIED CRYSTALLOGRAPHY

ISSN 1600-5767

Received 11 February 2016
Accepted 15 March 2016

Edited by A. R. Pearson, Universität Hamburg, Germany

Keywords: computer programs; *XDS*; *XDSAPP*; diffraction data processing.

Supporting information: this article has supporting information at journals.iucr.org/j

XDSAPP2.0

Karine M. Sparta,[a,b]* Michael Krug,[a,b] Udo Heinemann,[b,c] Uwe Mueller[a] and Manfred S. Weiss[a]*

[a]Macromolecular Crystallography, Helmholtz-Zentrum Berlin für Materialien und Energie, Albert-Einstein-Strasse 15, Berlin D-12489, Germany, [b]Macromolecular Structure and Interaction, Max Delbrück Center for Molecular Medicine, Robert-Rössle-Strasse 10, Berlin D-13125, Germany, and [c]Chemistry and Biochemistry Institute, Freie Universität Berlin, Takustrasse 6, Berlin D-14195, Germany. *Correspondence e-mail: karine.sparta@helmholtz-berlin.de, msweiss@helmholtz-berlin.de

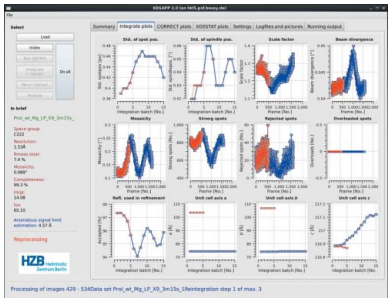
XDSAPP is an expert system and graphical user interface (GUI) for the automated processing of diffraction images using the *XDS* program suite and other programs. The latest major update and the extension of the program are presented here. The update includes new features, as well as improvements in the GUI and the underlying decision-making system. *XDSAPP* is freely available for academic users.

1. Introduction

In the past decade, macromolecular crystallography (MX) has experienced constant improvements in beamline efficiency at synchrotron sources around the world. In particular, the advent of modern hybrid photon-counting detectors has allowed the fast and shutterless collection of complete single-crystal X-ray diffraction data sets within a few minutes (Helliwell & Mitchell, 2015). This increases the necessity of providing users with a reliable and automated procedure to process all recorded diffraction images in a reasonable amount of time during or shortly after their measurement.

Popular software suites to handle such data include *iMOSFLM* (Leslie, 2006; Battye *et al.*, 2011), *HKL2000* (Otwinowski & Minor, 1997), *XDS* (Kabsch, 1993, 2010*a*,*b*) and *d*TREK* (Pflugrath, 1999). Amongst these, *XDS* is particularly suited to the direct processing of large diffraction data sets collected on beamlines, since it makes use of multi-processor hardware to speed up parallelizable calculations. Moreover, its three-dimensional profile-fitting procedure to estimate the intensities of reflections takes full advantage of the fine φ-slicing possibility offered by modern hybrid pixel detectors (Mueller, Wang & Schulze-Briese, 2012). Consequently, *XDS* is widely used to process diffraction data on synchrotron MX beamlines. A brief survey showed that it is installed and available on about 60% of all MX beamlines around the world. Nevertheless, despite all these advantages of *XDS* there are also a few serious disadvantages. Since *XDS* is a command-line-based program, it can be rather cumbersome to use in manual mode. In particular, the handling of lengthy input and output text files constitutes a significant hurdle for new and less-experienced users and is hence a potential source of error.

Over the years, several efforts have been reported to automate data processing in MX. These include the command-line expert system *ELVES* (Holton & Alber, 2004), the above-mentioned semi-automated *iMOSFLM* (Leslie, 2006; Battye *et al.*, 2011), *XIA2* (Winter, 2010; Winter *et al.*, 2013), *autoPROC* (Vonrhein *et al.*, 2011), *AutoProcess* (Grochulski *et al.*, 2012),



XDSme (P. Legrand; http://code.google.com/p/xdsme/) and the *DIALS* framework (Waterman *et al.*, 2013), as well as *XDSAPP* (Krug *et al.*, 2012) and *XDSi* (Kursula, 2004). Some systems extend beyond mere data processing and connect to the subsequent steps of automated phasing and model building, *e.g. ELVES*, *HKL3000* (Minor *et al.*, 2006), the *EDNA* framework and the Grenoble Automatic Data Processing System *GrenADES* developed at the ESRF (Monaco *et al.*, 2013).

XDSAPP (*XDS* automation and plotting protocols) was originally developed as a Tcl/Tk graphical user interface (GUI) for the automated use of *XDS* (Krug *et al.*, 2012). It constitutes a convenient interface to *XDS* and further relevant software needed for automated decision making, for instance for space-group selection. For user convenience, all important statistics from the *XDS* output files are represented graphically. Since February 2014, a completely new version of *XDSAPP* has been made available for download from the MX web page (http://www.helmholtz-berlin.de/bessy-mx) of the Helmholtz-Zentrum Berlin (HZB). The new *XDSAPP* GUI has been designed to provide users with a much simplified and more intuitive way of handling diffraction data sets. *XDSAPP* has been well adopted by the MX community, with over 500 research groups using it worldwide. Most *XDSAPP* users are from institutions located in Europe, but there are also significant numbers in North America and East Asia (Fig. 1). *XDSAPP* is constantly being adapted in response to changes in the third-party software used, and its functionality is being extended on the basis of users' feedback.

2. Methods

2.1. *XDSAPP* structure

For developing the new implementation of *XDSAPP2.0*, the interpreted language Python, pre-installed in most Linux-based operating system distributions, and *PyQt4*, a Python binding of the application framework *Qt*, were selected. The graphical plots are now generated using the *Qwt5* library. These all present the advantage of being free software published under the GNU General Public License and being portable across various operating systems. For licensing reasons, *PyQt4* and *Qwt5* are not distributed together with *XDSAPP*; they have to be obtained and installed separately by the user (see next section).

Figure 1
The geographic distribution of *XDSAPP* users around the world, as of January 2016. The figure was created using templates from Wikipedia (https://fr.wikipedia.org/wiki/Modèle:Géolocalisation/Monde).

2.2. Software environment

XDSAPP2.0 has been tested extensively on the following operating systems: Scientific Linux 6.3, Ubuntu 15.04 down to 12.04, openSUSE 13.1, and MacOS 10.9 (Mavericks) and 10.10 (Yosemite). It requires the installation of Python 2.7 or higher for the command-line version, as well as *PyQt4* (http://www.riverbankcomputing.com/software/pyqt) and *Qwt5* (http://qwt.sourceforge.net/) for the GUI. Essential for the processing of diffraction images is the installation of the latest version of *XDS* (Kabsch, 2010*a*,*b*). To take advantage of all *XDSAPP* features, the user is strongly recommended to install the program *XDSSTAT* (Diederichs, 2007), the *CCP4* suite (Collaborative Computer Project, Number 4, 1994; Winn *et al.*, 2011) and *PHENIX.XTRIAGE* (Adams *et al.*, 2010) for additional statistical analysis of the data sets, as well as *XDSVIEWER* (Hoffer, 2009) to visualize the images produced by *XDS* and *XDSSTAT*. Finally, the shell `tcsh` should be available on the user's computer system.

2.3. Hardware environment

In order to allow the fast processing of diffraction data sets on the fly at a beamline and to take full advantage of the highly parallelized steps in *XDS*, three servers are currently available for users of the HZB-MX beamlines: an HP DL580 G7 40 CPU-core server with 529 GB RAM (System 1) and, since the recent upgrade of beamline BL14.2 (Mueller, Darowski *et al.*, 2012; Mueller *et al.*, 2015), two HP DL580 Gen8 60 CPU-core servers with 258 GB RAM (System 2). Each server is connected to a detector server *via* a 10 Gb Ethernet point-to-point connection and is uplinked to a centralized 30 TB SAN storage array, providing an optimal environment to run the CPU-intensive *XDS* jobs while minimizing the network load.

2.4. GUI layout

The design of the new GUI for *XDSAPP* has been inspired by the ergonomic and appealing GUI for data processing developed for *iMOSFLM* (Leslie, 2006; Battye *et al.*, 2011). The main window of the *XDSAPP* GUI is divided into four parts (Fig. 2). The upper left-hand part titled 'Select' contains a command block with buttons to start jobs. The lower left part 'In brief' lists the most important statistics of the data set as processing progresses. More prominently, the upper right and largest part of the window is divided into seven tabs and a large screen window. Here, the user will find the result screens of the different processing steps and the form for modifying the processing parameters from their default values. Finally, the last part of the window is the status line at the bottom, which informs the user about the current job status.

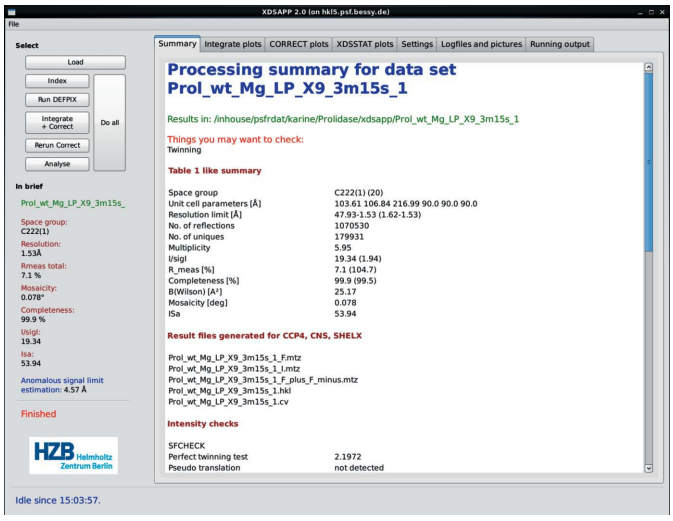
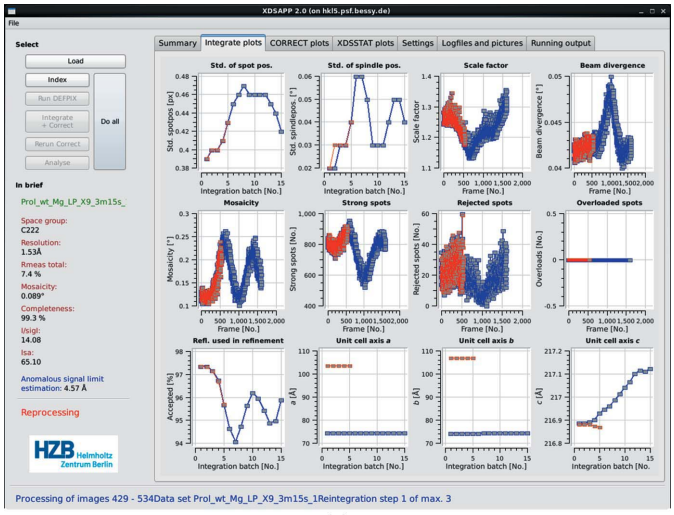
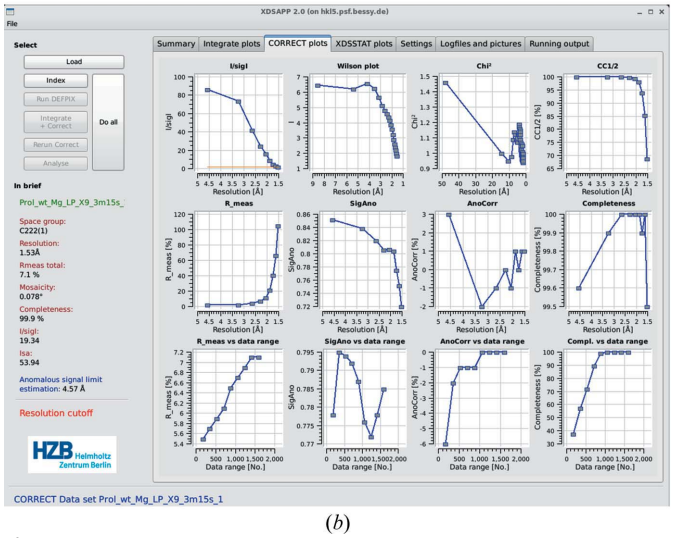


Figure 2
Organization of the *XDSAPP* GUI. The displayed 'Summary' tab shows the first part of the final processing summary for the prolidase test case, collected on the HZB beamline BL14.1.



(*a*)



(*b*)

Figure 3
'Integrate + CORRECT': real-time graphical output of the statistical tables in the *XDS* output files. (*a*) `INTEGRATE.LP` file, (*b*) `CORRECT.LP` file.

3. Usage

Unlike its Tcl/Tk predecessor (Krug *et al.*, 2012), *XDSAPP2.0* does not require a template file for each detector type, and its use is no longer restricted to the detectors available at the HZB-MX beamlines (Mueller, Darowski *et al.*, 2012; Mueller *et al.*, 2015). The important experimental parameters are read directly from the header of the diffraction images using the bash script *generate_XDS.INP* (Diederichs, 2015). In principle, all detectors supported by this script can be used with *XDSAPP*. However, only data from Dectris PILATUS 6M and Rayonix MX-225 detectors, which are currently in use on the HZB-MX beamlines BL14.1–3, have been tested extensively by the authors. New optional parameters have been introduced to allow the processing of data from diffractometers using different geometric definitions of the detector and rotation spindle axes.

Currently, *XDSAPP* offers two modes of operation: a GUI version, suitable for processing a single data set at a time and allowing varying degrees of control over the parameters used by *XDS*, and a command-line version to process automatically all data sets contained in a directory, with no user interaction after launching of the processing job.

3.1. GUI mode

Upon loading a data set, *XDSAPP* prepares the `XDS.INP` input file for *XDS* from the experimental information contained in the header of the image files, using the script *generate_XDS.INP* (Diederichs, 2015). Optimal default values for certain parameters, *e.g.* `SEPMIN` and `CLUSTER_RADIUS`, are selected according to the detector type.

Once a data set has been loaded using the upper left command block in *XDSAPP*, it can be processed either in a stepwise manner or completely automatically. For stepwise processing, the user first starts the autoindexing of the data set, examines the outcome, and then progresses through the integration and scaling steps. If the results are not satisfactory, the user can modify parameters in the 'Settings' tab and re-run the job. A typical user intervention here would be to change the parameters associated with `SPOT_RANGE` in order to base the indexing on more or fewer images. Using the 'Do all' button, a data set can also be fully processed using default parameters determined from the experimental settings.

3.1.1. Stepwise. The 'Indexing' step comprises the *XDS* steps *XYCORR*, *INIT*, *COLSPOT* and *IDXREF*. In some cases, indexing may fail, as shown by a low percentage of indexed diffraction spots and an obviously wrong refined detector distance deviating significantly from the experimental value given in the header of the images. Assuming that the detector distance read from the image header is correct, *XDSAPP* then restarts indexing without refining the distance automatically. If indexing still fails, the program alerts the user by issuing a pop-up message requiring the user to check the results carefully. If the diffraction images are indexed successfully, further processing can be performed.

By clicking the button 'DEFPIX', the determination of the trusted detector region is initiated and the resulting image

BKGPIX.cbf is displayed using *XDS-VIEWER* for visual inspection. The minimum value of the parameter VALUE_RANGE_FOR_TRUSTED_DETECTOR_PIXELS is set by *XDSAPP* depending on the detector used, but it may be changed according to the user's needs. If the user decides to change it, *DEFPIX* needs to be re-run.

'Integrate + CORRECT' first performs the integration of all reflections using the triclinic cell determined by *IDXREF*. Subsequently, *CORRECT* is run to determine the Bravais lattice and Laue group of the crystal. Further reintegration cycles are then performed based on the symmetry determination by *CORRECT*. During processing, the graphs for all relevant properties of the data sets are constantly updated, such as the evolution of the crystal mosaicity throughout the integration or the signal-to-noise ratio as a function of resolution determined by *CORRECT* (Fig. 3).

'Rerun CORRECT' can be used after the 'Integrate + CORRECT' procedure if the user wishes to modify parameters such as space group or resolution limit.

'Analyse' is the last step of *XDSAPP* and performs all statistical evaluation and data-conversion steps. *XDSSTAT* is run to obtain information about radiation damage during data collection (Diederichs, 2006), which is graphically represented in the tab '*XDSSTAT* plots' (Fig. 4). The script *XDSCONV* is used to produce intensity files suitable for *CCP4*, *SHELX* and *CNS*. In the case of ambiguous systematic extinction rules, files are created for all possible chiral space groups. *SFCHECK* (Vaguine *et al.*, 1999) and *PHENIX.XTRIAGE* provide information on possible pseudo-translation and twinning in the crystal. Finally, a summary of the processing is provided in HTML and ASCII result files, as well as in the 'Summary' tab of *XDSAPP* (Fig. 2).

3.1.2. 'Do all'. Clicking this button launches fully automated processing, consisting of the steps 'Indexing', 'DEFPIX', 'Integrate + CORRECT' and 'Analyse'. A *DEFPIX* run without graphical feedback is also included in the step 'Integrate + CORRECT'.

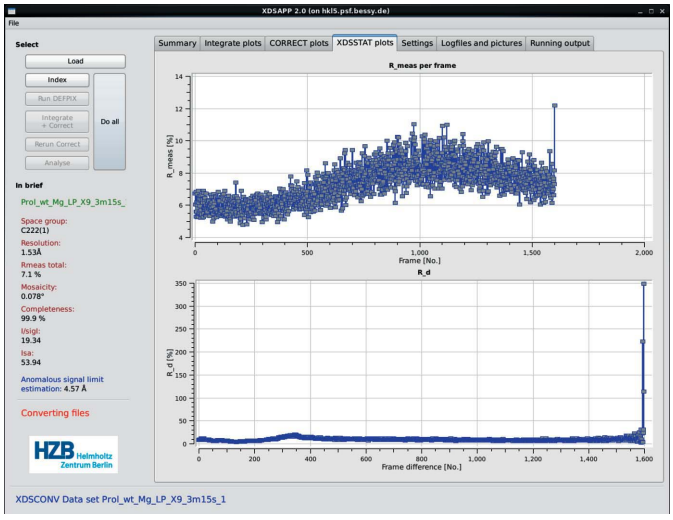


Figure 4
XDSSTAT plots.

3.1.3. Live processing. This mode allows the user to start processing a data set while it is still being collected. The user needs to provide the total number of images to be expected for the data set (parameter DATA_RANGE). *XDSAPP* divides the data set into four parts (or just two parts in the case of faster data collection with a PILATUS detector) and processes them in sequential runs, as soon as the images become available. *XYCORR* and *INIT* are only performed during the first processing run. The last run comprises all the diffraction images of the data set. Although the data in the first processing runs may be far from complete, these early results provide feedback concerning the quality of the crystal and/or data-collection strategy and may help the user to decide on the continuation or abortion of a lengthy data collection.

3.2. Command-line mode

3.2.1. Description. The command-line mode of *XDSAPP* offers the same functionalities as the GUI mode, using default values for data processing for single data sets. However, for historical reasons it differs in its implementation: the GUI is written using an object-oriented approach, while the command-line mode has a procedural structure. In the future, both modes will be combined and share the same object-oriented methods.

The command-line version of *XDSAPP* can be invoked from a shell by executing the command `xdsapp --cmd`. As in the 'Settings' tab of the GUI, it is possible to modify some parameters from their defaults. All available options can be listed using the command `xdsapp --help` and are detailed in Table S1 of the supporting information.

3.2.2. Multiple data sets. The command-line mode of *XDSAPP* with the option `--all` allows the sequential processing of all data sets present in a folder and all its subfolders. Combined with the continuous mode option `--continuous scan`, users at a beamline can start automated data processing during data collection from the top folder containing all measurements and let *XDSAPP* periodically look for new data sets, until the program is manually interrupted. All optional input is used for all data sets in the subdirectory tree found by *XDSAPP*. For example, the use of the option `--spacegroup` to provide a custom space group and cell parameters would be applied to all data sets and only makes sense in the context of an experiment in which all data sets are from the same type of crystal, *e.g.* a multi-crystal experiment or a fragment-screening experiment.

3.3. Different image names or new detector types

In its GUI mode, *XDSAPP2.0* lists for selection all data sets within a directory chosen by the user, based on the images it finds there. Images are defined as files with the extensions `.cbf`, `.img`, `.mar2300`, `.marccd`, `.mccd`, `.osc` or `.pck`. If a user wishes to process data sets consisting of other images, currently the only way to do this is to modify the code in the *XDSAPP* file `xdsit.py`, by adding a new file extension to the list in line 23. The correct definition of the detector axes, rotation spindle and polarization plane normal needs to be

checked carefully, either in the 'Settings' tab of the GUI or using the corresponding command-line options (Table S1). As mentioned above, the file `XDS.INP` is generated from the information contained in the header of the diffraction images, using the script *generate_XDS.INP* (Diederichs, 2015).

3.4. *XDSAPP* features on the HZB-MX beamlines

The beamline version of *XDSAPP* includes the latest stable development features of the program. This version contains options which are not yet in the release version. For instance, users from the small-molecule crystallography field on the HZB-MX beamlines have the possibility to output an additional intensity file named `xds.sad` containing the direction cosines of the reflections for subsequent absorption correction. If this option is checked, the orientation matrix of the reciprocal cell of the crystal is also given in the result files. The direction cosines are calculated during the 'Analyse' step from the `XDS_ASCII.HKL` file using the utility *XDS2SAD* (Sheldrick, 2008). Since there is only one binary file available for download, which has not been compiled for Ubuntu or MacOS, this feature has not been included in the release version of *XDSAPP* to ensure the consistency of the program under different platforms.

Another example is that, in the command-line version, users also have the possibility of invoking initial structure refinement cycles using *PHENIX.REFINE* (Adams *et al.*, 2010) using a specific PDB model with the option `--refine`. Upon completion, *XDSAPP* uses the *CCP4* programs *FFT*, *MAPMASK* and *PEAKMAX* to produce electron-density maps suitable for visualization in *COOT* (Emsley *et al.*, 2010) and peak search.

Finally, at the end of processing, the results are stored in a local user database. Upon identification in a web form, users can access all their processed data sets. For each entry, a summary of the data set statistics is provided, as well as images of the *INTEGRATE* and *CORRECT* plots.

3.5. *XDSAPP* problem reporting and feedback

XDSAPP users who encounter a problem or identify a bug in the program, or who simply wish to comment on the program performance, may contact the developers by email at xdsapp@helmholtz-berlin.de. Suggestions for new features may also be communicated to the developers in this way. For bug reports, the user is recommended to send an accurate description of the problem and a copy of the terminal output containing a possible error message. *XDSAPP* also creates a hidden file named `.xdsapp` in the output folder. Users reporting a problem should search for this file, and, if it is present, attach it to the bug report.

4. Decision-making in *XDSAPP*

XDSAPP does not just provide a GUI for *XDS*; it is first and foremost an expert system for the processing of diffraction images using *XDS*, relying on several automated decision-making steps. In the following, the most important decision points are discussed in some detail.

4.1. Space-group selection

At the beginning of data processing, no assumption is made about the symmetry of the crystal. The space group *P*1 is used for the first integration and subsequent *CORRECT* run. Since no analysis of systematic extinctions for screw axes is made by *CORRECT*, the space group with the lowest number corresponding to the Bravais lattice and Laue group determined by *CORRECT* is used in the next smart reintegration cycles. The reflection list is then analysed by *POINTLESS*, and a list of possible space groups is output, sorted by their probabilities. The space group with the highest probability is selected for the final *CORRECT* run and analysis of the data set. In the case of enantiomorphic space groups with equal probability, all relevant output files are created for each space group.

4.2. Smart reintegration cycles

The aforementioned smart reintegration cycles, performed after the first integration run in *P*1, ensure that a data set with the best possible statistical properties is produced from the diffraction images. By default, a maximum of three integration cycles is performed. In each cycle, *INTEGRATE* and *CORRECT* are run consecutively. The cell parameters, mosaicity and orientation of the crystal and the direct-beam direction are refined and updated, as well as the direction of the rotation axis in *CORRECT*. Next, *XDSAPP* compares the $R_{\rm meas}$ values (Einspahr & Weiss, 2012) in the file `CORRECT.LP` with those from the previous run. If the improvement is less than a given threshold, the smart reintegration cycles are interrupted; otherwise *XDSAPP* starts a new cycle. In a future version of *XDSAPP*, improvements in data quality will be gauged by monitoring the asymptotic value of $I/\sigma(I)$ (ISa) of a data set (Diederichs, 2010) or the half data set correlation coefficient $CC_{1/2}$, instead of $R_{\rm meas}$.

4.3. Resolution limit

The resolution cutoff of a data set is performed at the end of the 'Integrate + CORRECT' procedure. Three parameters are taken into account for the estimation of the resolution limit: the signal-to-noise ratio, the completeness and $R_{\rm meas}$ in the last resolution shell, as read from the `CORRECT.LP` file. From our experience, the signal-to-noise ratio in the last resolution shell is usually the parameter playing the most important role in the resolution limit estimation. The current beamline version of *XDSAPP* uses an iterative approach after each reintegration cycle for a reliable estimation of the resolution limit. Given the current discussions and developments in the field (Karplus & Diederichs, 2012, 2015), a future version of *XDSAPP* will make use of more objective data quality indicators, such as the half data set correlation coefficient $CC_{1/2}$.

4.4. Detection of anomalous signal

Since the latest *XDSAPP* release, a robust and conservative procedure has been implemented to identify automatically the

presence or absence of anomalous signal in a data set. The anomalous signal is checked from the file CORRECT.LP after the first integration in space group *P*1, and *CORRECT* is run with the parameter FRIEDEL'S_LAW set to FALSE. This parameter is changed if no anomalous signal is found. Its default value in *XDSAPP* is now UNKNOWN instead of FALSE; if the user sets it to TRUE or FALSE before processing, this choice is retained and no detection of anomalous signal is performed.

The first *CORRECT* run is performed with the parameter STRICT_ABSORPTION_CORRECTION set to FALSE. In the case of anomalous data, this parameter is only set to TRUE for the following runs if one of the three χ^2 values of fit of the correction factors in the file CORRECT.LP is higher than 1.5, which is a more relaxed criterion than the recommendation given in the *XDS* wiki pages (Diederichs, 2014).

5. Performance

5.1. Prolidase

The first test case is a data set from a native prolidase crystal collected on the HZB-MX beamline BL14.1 (Mueller, Darowski *et al.*, 2012; Mueller *et al.*, 2015), containing 1600 diffraction images of 0.1° rotation each recorded on a PILATUS 6M detector (Table 1). It was used to compare the time performance of the 'Do all' procedure of the *XDSAPP* GUI with three reintegration cycles on different 64 bit systems: our beamline System 1 and System 2 servers, a personal computer (PC) operating Ubuntu 15.04 with 4 × 3.10 GHz Intel Core i5 CPUs and 16 GB RAM, and a MacBookPro 6.2 notebook running OSX Yosemite 10.10.5 with 2 × 2.66 GHz Intel Core i7 CPUs and 4 GB RAM (Table 2). For comparison, the same resolution limit and spot ranges for *COLSPOT* were fixed for all runs. Since *XDSAPP2.0* was released before the *XDS* version of 15 October 2015, no fine-tuning of the parameter NUMBER_OF_IMAGES_IN_CACHE was performed to improve the time performance for this test case, and the default value of 101 was used. *XDSAPP* needs about eight times less runtime on our System 2 server than on a notebook. Processing times of about 10 min correspond to the duration of an average full data collection on our beamlines, allowing users to obtain results without delay during their measurements.

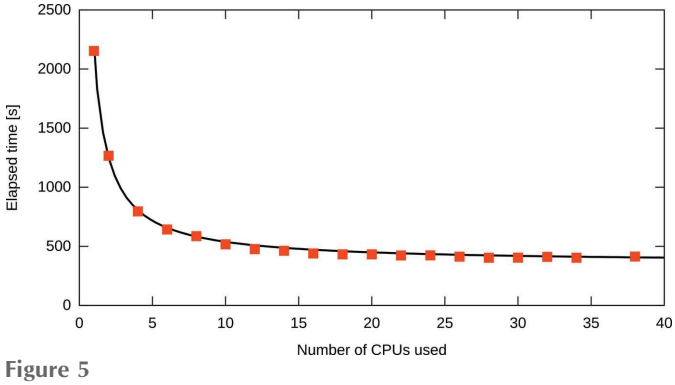


Figure 5
XDSAPP benchmark tests on a System 2 server: elapsed time as a function of the number of CPUs used. The black line represents the fit of equation (1).

Table 1
Data collection and processing statistics of the native prolidase data set.

Data set	Native prolidase
X-ray source (beamline)	BESSY II (BL14.1)
Detector	PILATUS 6M
Temperature (K)	100
Wavelength (Å)	0.918
Crystal-to-detector distance (mm)	283.6
Rotation range per image (°)	0.1
Total rotation range (°)	160
Exposure time per image (s)	0.65
Space group	$C222_1$
Resolution range (outer shell) (Å)	47.93–1.53 (1.62–1.53)
Unit-cell parameters *a*, *b*, *c* (Å)	103.6, 106.8, 217.0
Mosaicity (°)	0.078
Total No. of reflections	1 070 530
Unique reflections	179 931
Multiplicity	5.95
$\langle I/\sigma(I)\rangle$ (outer shell)	19.34 (1.94)
ISa	53.9
Completeness (outer shell) (%)	99.9 (99.5)
R_{meas} (outer shell) (%)	7.1 (104.7)
Overall *B* factor from Wilson plot (Å^2)	25.2

Table 2
Time performance of *XDSAPP*: comparison of the time needed for the 'Do all' procedure in the GUI for different computing environments.

Platform	Notebook	PC	Server 1	Server 2
Time	1 h 11 min 45 s	26 min 47 s	12 min 47 s	7 min 19 s

By default, *XDSAPP* uses all available CPUs for *XDS* jobs, with MAXIMUM_NUMBER_OF_JOBS set to 3. However, the situation at the HZB-MX beamlines is different. Benchmark tests on a System 2 server were performed using the *XDS* version of 15 October 2015 (Fig. 5). The prolidase data set was stored locally on the server, and subsequent *XDSAPP* full processing jobs with varying numbers of CPUs were launched from a script. Between each *XDSAPP* job, the output folder was deleted and the cache emptied. The elapsed time as a function of the number of CPUs used is well fitted by the function for theoretical runtime derived from Amdahl's law (Amdahl, 1967):

$$T(n) = (1 - p + p/n)T(1), \quad (1)$$

where $T(1)$ is the total runtime for one CPU, equal in this example to 2153 s, n is the number of CPUs used and p is the fraction of parallelized tasks, refined to 0.879 (2). The elapsed time does not decrease substantially when using more than 16 CPUs (439 s). Hence, this value has been implemented as the default for data processing on the HZB-MX beamlines.

5.2. Multiple data sets

The second test case is a set of 69 endothiapepsin data sets collected on BL14.1 as part of a fragment-screening campaign. The data sets were processed sequentially using the command-line mode of *XDSAPP* for multiple data sets with 16 CPUs per

Table 3
XDSAPP processing results for the 69 endothiapepsin data sets.

X-ray source (beamline)	BESSY II (BL14.1)
Detector	PILATUS 6M
Temperature (K)	100
Wavelength (Å)	0.918
Crystal-to-detector distance (mm)	151.1–476.5
Rotation range per image (°)	0.1
Total rotation range (°)	180–200
Space group	$P2_1$
Lower resolution limit (Å)	42.94–36.81
Higher resolution limit (Å)	2.07–1.15
Unit-cell parameters a, b, c, β (Å, °)	45.1–45.6, 72.4–74.3, 52.1–54.0, 108.8–110.8
Mosaicity (°)	0.593–0.063
Total No. of reflections	67 711–421 927
Unique reflections	20 469–113 822
Multiplicity	3.13–3.92
$\langle I/\sigma(I)\rangle$ (outer shell)	6.0–34.3 (1.8–19.7)
ISa	15.0–69.4
Completeness (outer shell) (%)	90.2–99.6 (73.0–99.5)
R_{meas} (outer shell) (%)	3.3–21.3 (6.6–79.5)
Overall B factor from Wilson plot (Å^2)	15.4–23.8

job on a System 2 server (Table 3). Since no change in space group and unit-cell parameters was expected, all data sets were processed with the respective values from the ligand-free structure (Köster *et al.*, 2011) using the option `--spacegroup`. In total, *XDSAPP* needed 3 h 20 min and 48 s for all data sets. The processing times for individual data sets ranged from 102 to 256 s (Fig. 6). If pre-refinement cycles with *PHENIX.REFINE* were included, the total elapsed time was 8 h 30 min and 54 s, with individual times ranging from 312 to 644 s, depending on the resolution of the data set.

6. Conclusions

Since its first release, *XDSAPP* has been completely re-implemented using the interpreted language Python. More visible to its users, its graphical user interface has been rewritten with *PyQt4*, giving users a more intuitive feeling than Tcl/Tk, since *Qt* uses the native application programming interface of the operating system it is used on.

Within the GUI, *XDS* tasks have been separated to allow either fully automated processing or the use of a step-by-step procedure. Behind the scenes, more decision-making steps have been implemented, such as, for instance, the detection of anomalous signal and the setting of related parameters for *CORRECT*.

Thanks to extensive testing on the latest versions of Linux distributions and MacOS, *XDSAPP* has evolved into a cross-platform program. Moreover, its stability has been improved by thorough exception handling.

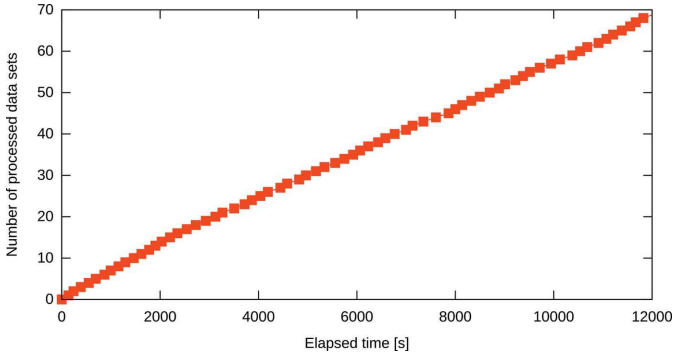


Figure 6
XDSAPP for multiple data sets: number of processed data sets as a function of elapsed time.

7. Outlook

For future versions of *XDSAPP*, we are planning to simplify the program further by using only an object-oriented approach. This should lead to significantly more stable and less error-prone code. The core of the program will be decoupled from the user interface, which will allow the easy use of *XDSAPP* from a browser, for example. Concerning the GUI, the use of *Qwt5* will have to be discontinued, since *Qwt5* is no longer supported and hardly installable in the newest OS like RHEL. A possible option for replacement of *Qwt5* could be *matplotlib* (http://matplotlib.org/).

Concerning processing, current developments aim to improve the estimation of the resolution cutoff and reduce the processing time for live processing on synchrotron beamlines. *XDSAPP* will become more flexible to allow the use of detector-specific parameters and correction files. It should be possible to load defined geometry parameters for all known MX beamlines worldwide. An important feature to be implemented is the manual selection, with the mouse, of untrusted detector regions for *DEFPIX* by interacting with the file `BKGPIX.cbf` or the diffraction images, as is possible in *XDSGUI* (Brehm *et al.*, 2015). Also, more checks will be implemented to detect processing failures early and to use different strategies to overcome them.

Finally, a feedback button for problem reporting or communication with the developers will be implemented directly in the GUI.

Acknowledgements

The authors gratefully acknowledge financial support from the Joint Berlin MX Laboratory and the BMBF *via* the Röntgen–Ångström Cluster (project No. 05K13CB1). We thank Michael Hellmig (HZB) for IT support and Kay Diederichs (University of Konstanz, Germany) for many helpful discussions. The development of *XDSAPP* also owes a lot to the community of users at the HZB-MX beamlines BL14.1–3 for testing and valuable feedback. Dr Peter J. Stogios (University of Toronto) and Dr Vito Calderone (University of Florence) have been very helpful in providing test data sets from different detectors from those available at our beamlines. Finally, we thank Dr Piotr Wilk (Humboldt University of Berlin and HZB) for the provision of the prolidase data set and Dr Franziska Huschmann (University of Marburg and HZB) for the endothiapepsin data sets.

References

Adams, P. D. *et al.* (2010). *Acta Cryst.* D**66**, 213–221.

Amdahl, G. M. (1967). *AFIPS Spring Joint Computer Conference*, 18–20 April 1967, Atlantic City, New Jersey, USA, *AFIPS Conference Proceedings* Vol. 30, pp. 483–485. Washington DC: Thompson Books.
Battye, T. G. G., Kontogiannis, L., Johnson, O., Powell, H. R. & Leslie, A. G. W. (2011). *Acta Cryst.* D**67**, 271–281.
Brehm, W., Diederichs, K. & Hoffer, M. (2015). *XDSGUI*. Version of 17 September 2015. https://sourceforge.net/projects/xdsgui/.
Collaborative Computational Project, Number 4 (1994). *Acta Cryst.* D**50**, 760–763.
Diederichs, K. (2006). *Acta Cryst.* D**62**, 96–101.
Diederichs, K. (2007). *XDSSTAT*, http://strucbio.biologie.uni-konstanz.de/xdswiki/index.php/Xdsstat.
Diederichs, K. (2010). *Acta Cryst.* D**66**, 733–740.
Diederichs, K. *et al.* (2014). *Tips and Tricks*. Retrieved from http://strucbio.biologie.uni-konstanz.de/xdswiki/index.php/Tips_and_Tricks on 5 May 2014.
Diederichs, K. *et al.* (2015). *Generate_XDS.INP (script)*. Retrieved from http://strucbio.biologie.uni-konstanz.de/xdswiki/index.php/Generate_XDS.INP on 2 March 2015.
Einspahr, H. M. & Weiss, M. S. (2012). *International Tables for Crystallography*, Vol. F, *Crystallography of Biological Macromolecules*, 2nd ed., ch. 2.2, pp. 64–74. Heidelberg: Springer.
Emsley, P., Lohkamp, B., Scott, W. G. & Cowtan, K. (2010). *Acta Cryst.* D**66**, 486–501.
Grochulski, P., Fodje, M., Labiuk, S., Gorin, J., Janzen, K. & Berg, R. (2012). *J. Struct. Funct. Genomics*, **13**, 49–55.
Helliwell, J. R. & Mitchell, E. P. (2015). *IUCrJ*, **2**, 283–291.
Hoffer, M. (2009). *XDS-Viewer*, http://sourceforge.net/projects/xds-viewer.
Holton, J. & Alber, T. (2004). *Proc. Natl Acad. Sci. USA*, **101**, 1537–1542.
Kabsch, W. (1993). *J. Appl. Cryst.* **26**, 795–800.
Kabsch, W. (2010*a*). *Acta Cryst.* D**66**, 125–132.
Kabsch, W. (2010*b*). *Acta Cryst.* D**66**, 133–144.
Karplus, P. A. & Diederichs, K. (2012). *Science*, **336**, 1030–1033.
Karplus, P. A. & Diederichs, K. (2015). *Curr. Opin. Struct. Biol.* **34**, 60–68.
Köster, H., Craan, T., Brass, S., Herhaus, C., Zentgraf, M., Neumann, L., Heine, A. & Klebe, G. (2011). *J. Med. Chem.* **54**, 7784–7796.
Krug, M., Weiss, M. S., Heinemann, U. & Mueller, U. (2012). *J. Appl. Cryst.* **45**, 568–572.
Kursula, P. (2004). *J. Appl. Cryst.* **37**, 347–348.
Leslie, A. G. W. (2006). *Acta Cryst.* D**62**, 48–57.
Minor, W., Cymborowski, M., Otwinowski, Z. & Chruszcz, M. (2006). *Acta Cryst.* D**62**, 859–866.
Monaco, S., Gordon, E., Bowler, M. W., Delagenière, S., Guijarro, M., Spruce, D., Svensson, O., McSweeney, S. M., McCarthy, A. A., Leonard, G. & Nanao, M. H. (2013). *J. Appl. Cryst.* **46**, 804–810.
Mueller, U., Darowski, N., Fuchs, M. R., Förster, R., Hellmig, M., Paithankar, K. S., Pühringer, S., Steffien, M., Zocher, G. & Weiss, M. S. (2012). *J. Synchrotron Rad.* **19**, 442–449.
Mueller, U., Förster, R., Hellmig, M., Huschmann, F. U., Kastner, A., Malecki, P., Pühringer, S., Röwer, M., Sparta, K., Steffien, M., Ühlein, M., Wilk, P. & Weiss, M. S. (2015). *Eur. Phys. J. Pus*, **130**, 141–150.
Mueller, M., Wang, M. & Schulze-Briese, C. (2012). *Acta Cryst.* D**68**, 42–56.
Otwinowski, Z. & Minor, W. (1997). *Methods Enzymol.* **276**, 307–326.
Pflugrath, J. W. (1999). *Acta Cryst.* D**55**, 1718–1725.
Sheldrick, G. M. (2008). *XDS2SAD*. Version 2008/1. University of Göttingen, Germany.
Vaguine, A. A., Richelle, J. & Wodak, S. J. (1999). *Acta Cryst.* D**55**, 191–205.
Vonrhein, C., Flensburg, C., Keller, P., Sharff, A., Smart, O., Paciorek, W., Womack, T. & Bricogne, G. (2011). *Acta Cryst.* D**67**, 293–302.
Waterman, D. G., Winter, G., Parkhurst, J. M., Fuentes-Montero, L., Hattne, J., Brewster, A., Sauter, N. K. & Evans, G. (2013). *CCP4 Newsl. Protein Crystallogr.* **49**, 16–19.
Winn, M. D. *et al.* (2011). *Acta Cryst.* D**67**, 235–242.
Winter, G. (2010). *J. Appl. Cryst.* **43**, 186–190.
Winter, G., Lobley, C. M. C. & Prince, S. M. (2013). *Acta Cryst.* D**69**, 1260–1273.