Principles of data processing with XDS and XDSGUI

Kay Diederichs

Protein Crystallography /
Molecular Bioinformatics
University of Konstanz, Germany

Outline

- General information about *XDS*
- Usage, problems, diagnostics
- Demonstration of *XDSGUI:* processing of **your** data

throughout this talk: *program*, file

The *XDS* program suite

Original author:
Wolfgang Kabsch
(Max-Planck-Institute
Heidelberg)

Since ~1986

I joined in 2007

The *XDS* program suite

- *XDS*: the main program (indexing, integrating, scaling)
- *XSCALE*: scale several *XDS* intensity data sets together; zero-dose extrapolation; statistics
- *XDSCONV*: convert to other programs' formats

The following programs are independent of the *XDS* distribution:

- *XDS-Viewer* - inspect diagnostic images written by *XDS*, or (single) data frames (open source: sourceforge.net).Instead, *adxv* may be used
- *XDSSTAT, XDSCC12* - additional statistics (not part of main distribution; download and use: see XDSwiki)
- *XDSGUI* – graphical user interface (open source: sourceforge.net)

Distribution for 64bit Linux & Mac: latest version: Nov-2016 (w/ last error correction build Dec-2016); http://xds.mpimf-heidelberg.mpg.de/;
see XDSwiki: Installation

interfaces with ...

- beamline software (generating XDS.INP)
- scripts: *xia2* (CCP4), *autoPROC* (Globalphasing), *xdsme* (Soleil), *autoxds* (SSRL), *autoprocess* (CMCF), fastDP (Diamond) ... *generate_XDS.INP* (XDSwiki)
- CCP4: *pointless, xdsconv* (type CCP4, or CCP4_I, or CCP4_F)

Sources of information

- *XDS* main website: http://xds.mpimf-heidelberg.mpg.de - complete, accurate, up-to-date documentation; download

- XDSwiki: http://strucbio.biologie.uni-konstanz.de/xdswiki/index.php/Main_Page

- CCP4 bulletin board

- “XDS webinar” (http://www.rigaku.com/downloads/webinars/kay-diederichs/)

- “X-ray tutorial” (Faust *et al.* JAC 2008, 2010)

- Email to kay.diederichs@uni-konstanz.de

*XDS*wiki

- started Feb 2008; ~ 60 pages at
http://strucbio.biologie.uni-konstanz.de/xdswiki/index.php/Main_Page
- e.g. „Optimization“; explanations of task output
- „Tips and Tricks“, „FAQ“
- „Quality Control“ with datasets and results, and links to the projects of the ACA2011 and ACA2014 „data processing“ workshop
- anybody can contribute!
(same holds for CCP4wiki: ~ 90 pages at
http://strucbio.biologie.uni-konstanz.de/ccp4wiki/index.php/Main_Page)

XDS philosophy

(just a short selection)

- Do very little, but do it very well
- Very clearly structured
- Very robust – small molecule to ribosome

XDS features

(just a short selection)

- 3D - profiles of reflections are transformed into their own coordinate systems which makes them highly similar (Kabsch 1988 *J. Appl. Cryst*. **21**, 916-924)
- Smooth scaling (*ibid*.)
- Zero-dose extrapolation (*XSCALE*) can help a lot in sub-structure determination (Diederichs *et al.* 2003, *Acta Cryst.* **D59**, 903-909.)
- Fast - two levels of parallelization

XDS non-features

- Old-fashioned: ASCII output to files, graphics, no mouse-over „help“ bubbles
- Nothing automatic, user is in full control
- No frame header reading
- Incomplete space-group determination: screw axes not automatic
- No supporting organization, XDS workshops, advertising, funding, income
- No source code available (but papers document features thoroughly)

How to use *XDS* ?

- XDS needs a single input file XDS.INP with parameters describing data reduction
- Keywords and their parameters have the form e.g. DETECTOR_DISTANCE= 120.
- There are about 30 relevant keywords, but only about 15 are required (and may change between projects). All parameters have reasonable defaults where possible.
- shortcut: *generate_XDS.INP* from XDSwiki
- Run *xds_par* (on the commandline)

Example for MarCCD

```
JOB= XYCORR INIT COLSPOT IDXREF DEFPIX INTEGRATE CORRECT
ORGX=1546 ORGY=1552        !Detector origin (pixels); e.g. NX/2 NY/2
DETECTOR_DISTANCE=180      !(mm)
OSCILLATION_RANGE=0.50     !degrees (>0)
X-RAY_WAVELENGTH=0.980243 !Angstroem
NAME_TEMPLATE_OF_DATA_FRAMES=frms/wga2-27_1_???.img
DATA_RANGE=1 360           !Numbers of first and last data image collected
BACKGROUND_RANGE=1 10      !Numbers of first and last data image for background
SPACE_GROUP_NUMBER=  19    !0 for unknown crystals; cell constants are ignored.
UNIT_CELL_CONSTANTS= 44.4   86.4   104.5  90 90  90  ! not required if spgr=0
REFINE(IDXREF)=BEAM AXIS ORIENTATION CELL DISTANCE
REFINE(INTEGRATE)=DISTANCE BEAM ORIENTATION CELL ! AXIS
ROTATION_AXIS= 1.0 0.0 0.0
INCIDENT_BEAM_DIRECTION=0.0 0.0 1.0
FRACTION_OF_POLARIZATION=0.99                    ! SLS X06SA
POLARIZATION_PLANE_NORMAL= 0.0 1.0 0.0
DETECTOR=CCDCHESS        MINIMUM_VALID_PIXEL_VALUE=1     OVERLOAD=65000
DIRECTION_OF_DETECTOR_X-AXIS= 1.0 0.0 0.0
DIRECTION_OF_DETECTOR_Y-AXIS= 0.0 1.0 0.0
VALUE_RANGE_FOR_TRUSTED_DETECTOR_PIXELS= 7000 30000 !Used by DEFPIX
                                  !for excluding shaded parts of the detector.
INCLUDE_RESOLUTION_RANGE=50.0 1.3 !Angstroem; used by DEFPIX,INTEGRATE,CORRECT
```

Bold keyword/parameter pairs are required. Complete documentation at http://xds.mpimf-heidelberg.mpg.de/html_doc/xds_parameters.html

Templates for many detectors at http://xds.mpimf-heidelberg.mpg.de/html_doc/detectors.html

Principle of *XDS* processing

- *The basic idea is simple*
- There is one JOB= line in XDS.INP which specifies a list of tasks/jobs:

```
JOB= XYCORR INIT COLSPOT IDXREF DEFPIX INTEGRATE CORRECT
```

- data reduction is divided into tasks/jobs in **modular** way
- information storage/exchange/flow between tasks by data files which may be inspected/analyzed
- each task needs the result from the previous tasks
- fine-tuning of a task does *not* require previous tasks to be repeated
- each task writes its output file <TASK>.LP

Using *XDS* – steps („JOBS“)

- XYCORR : write positional correction files (X-CORRECTIONS.cbf, Y-CORRECTIONS.cbf)
- INIT : find background pixels (defaults usually OK)
- COLSPOT: find reflection positions
- IDXREF : "index" reflections; user may supply/choose spacegroup
- XPLAN [not required] : strategy for data collection
- DEFPIX : find beamstop shadow (defaults mostly OK)
- INTEGRATE : evaluates intensities on all frames, writes INTEGRATE.HKL and FRAME.cbf
- CORRECT : scales, rejects outliers, statistics, writes XDS_ASCII.HKL (and other files)

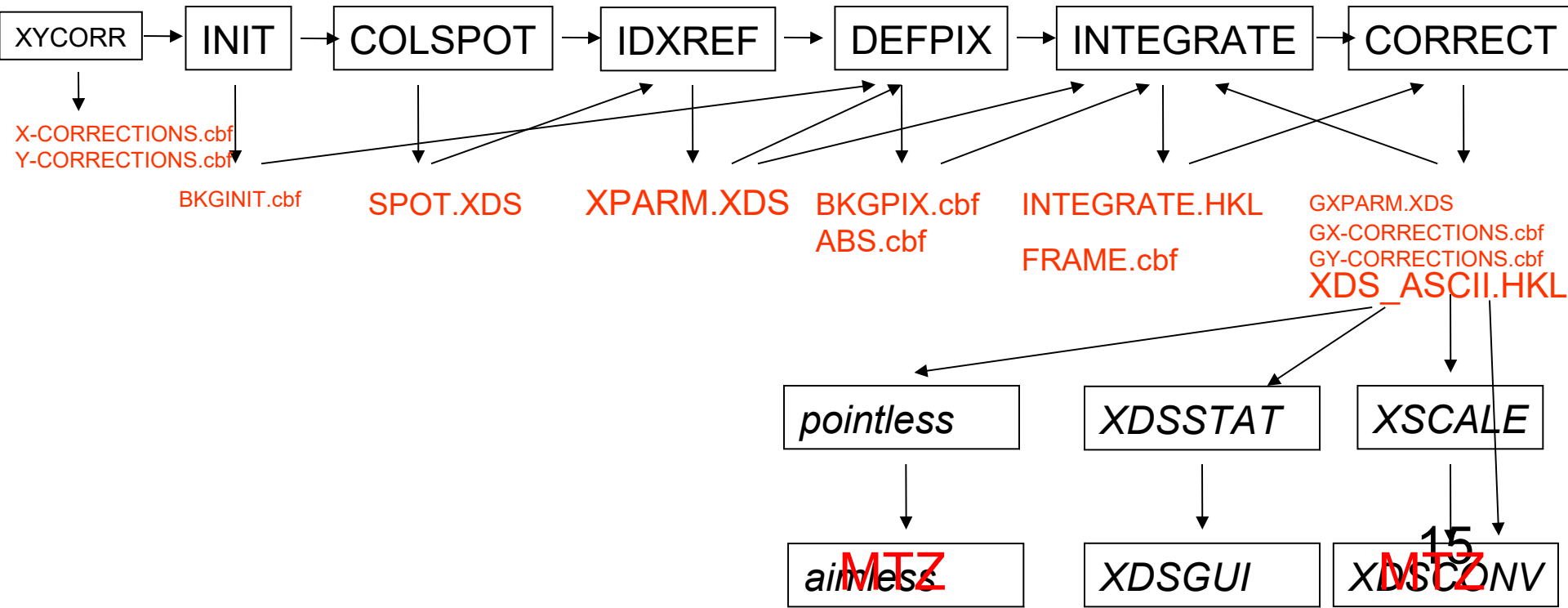
Information flow

NAME_TEMPLATE_OF_DATA_FRAMES
DETECTOR

OSCILLATION_RANGE
SEPMIN
STRONG_PIXEL

ORGX
ORGY
DETECTOR_DISTANCE
X_RAY_WAVELENGTH
SPACE_GROUP_NUMBER

DATA_RANGE



```
!FORMAT=XDS_ASCII    MERGE=FALSE    FRIEDEL'S_LAW=TRUE
!OUTPUT_FILE=XDS_ASCII.HKL        DATE= 3-Oct-2006
!Generated by CORRECT   (XDS VERSION  August 18, 2006)
!PROFILE_FITTING= TRUE
!SPACE_GROUP_NUMBER=   92
!UNIT_CELL_CONSTANTS=    57.71    57.71   150.08  90.000  90.000  90.000
!NAME_TEMPLATE_OF_DATA_FRAMES= ../series_2_????.img
!DATA_RANGE=       1     399
!X-RAY_WAVELENGTH=  0.939010
!INCIDENT_BEAM_DIRECTION=  0.001872 -0.002230  1.064947
!FRACTION_OF_POLARIZATION=   0.980
!POLARIZATION_PLANE_NORMAL=  0.000000  1.000000  0.000000
!ROTATION_AXIS=  0.999995  0.002477 -0.001917
!OSCILLATION_RANGE=  0.500000
!STARTING_ANGLE=    30.000
!STARTING_FRAME=       1
!DETECTOR=ADSC
!DIRECTION_OF_DETECTOR_X-AXIS=   1.00000   0.00000   0.00000
!DIRECTION_OF_DETECTOR_Y-AXIS=   0.00000   1.00000   0.00000
!DETECTOR_DISTANCE=   189.286
!ORGX=   1541.25  ORGY=   1535.30
!NX=  3072  NY=  3072    QX=  0.102600  QY=  0.102600
!NUMBER_OF_ITEMS_IN_EACH_DATA_RECORD=12
!ITEM_H=1
!ITEM_K=2
!ITEM_L=3
!ITEM_IOBS=4
!ITEM_SIGMA(IOBS)=5
!ITEM_XD=6
!ITEM_YD=7
!ITEM_ZD=8
!ITEM_RLP=9
!ITEM_PEAK=10
!ITEM_CORR=11
!ITEM_PSI=12
!END_OF_HEADER
     0     0     4  4.287E-01  2.814E-01  1501.6  1514.4     99.4 0.00920 100  27   75.39
     0     0    -4  2.243E-01  2.386E-01  1587.4  1548.6     91.6 0.00920 100  30  -79.02
     0     0     5  5.976E-03  3.443E-01  1490.9  1510.2    100.4 0.01150 100  22   74.94
```

XDS output file: XDS_ASCII.HKL

non-obvious

How do random and systematic *error* depend on the *signal*?

random error obeys *Poisson statistics*
error = square root of signal

Systematic error is *proportional* to signal
error = x * signal (e.g. x=0.02 ... 0.10)

(which is why James Holton calls it „fractional error“; there are exceptions)

non-obvious

Systematic errors (noise)

- beam flicker (instability) in flux or direction
- shutter jitter
- vibration due to cryo stream
- split reflections, secondary lattice(s), ice
- absorption from crystal and loop
- radiation damage
- detector calibration and inhomogeneity; overload
- shadows on detector
- deadtime in shutterless mode
- imperfect assumptions about the experiment and its geometric parameters in the processing software
- ...

The “error model”

Random error: $\sigma_r(I) \approx \sqrt{I}$

this is what INTEGRATE calculates

Systematic errors: $\sigma_s(I) \approx x*I$

this leads to deviations > $\sigma_r(I)$ between sym-related reflections

New $\sigma(I)$ estimate: $\sigma(I) = \sqrt{(a*(\sigma_r(I)^2+b*I^2))}$

with constants a,b fitted by CORRECT for the dataset

When random error vanishes (“asymptotically”), this results in $I/\sigma(I) = 1/\sqrt{(a*b)}$

A *proxy* for good data

$(I/sigma)_{asymptotic}$=ISa (reported in CORRECT.LP) is a measure of systematic error arising from beamline, crystal, and data processing

For a given data set, ISa increases if errors in the geometric parameterization are removed and e.g. the correct choice of “FRIEDEL'S_LAW=TRUE” versus “FALSE” is made. In short: when the experimental data are well processed

Maximizing ISa (good values are 30 and higher) *means minimizing systematic errors;*

This usually also optimizes $CC_{1/2}$ *at high resolution*

XDSCC12

$CC_{1/2}$ is a robust indicator of data precision

Q: How does $CC_{1/2}$ *change* when adding specific data?

A: calculate $CC_{1/2}$ with and without the data: difference is $\Delta CC_{1/2}$

* XDS_ASCII.HKL: a batch is (e.g.) 1° of data (use -t option!)

* XSCALE.HKL: each dataset is a batch

Useful to look at

* as a function of batch number and resolution (radiation damage?)

* separately for isomorphous and anomalous signal

Easy access: XDSGUI; download and documentation: XDSwiki

Assmann, G., Brehm, W. and Diederichs, K. (2016) Identification of rogue datasets in serial crystallography (2016) *J. Appl. Cryst.* **49**, 1021-1028.

XDSGUI

- Simple GUI using Qt
- Adapted to the XDS philosophy
- User – extensible / modifiable commands
- Plots synchronously while processing
- Documentation and availability: XDSwiki

Getting the best data

1) data processing is very logical in principle but the devil may be in the detail; look at the program output and plots!

2) As with any other step, it may be considered as *iterative*

3) Data processing stats tell *precision* only

4) Refinement R and CC values tell about *accuracy* of data and model

5) If unsure about details, try alternatives and decide based on refinement – but then be careful to compare the same reflections!

References

- Kabsch, W. (2010) *XDS*. *Acta Cryst.* **D66**, 125-132 (open access)
- Kabsch, W. (2010) Integration, scaling, space-group assignment and post-refinement. *Acta Cryst.* **D66**, 133-144 (open access)
- Diederichs, K., McSweeney, S., Ravelli, R. (2003) Zero-dose extrapolation as part of macromolecular synchrotron data reduction. *Acta Cryst* **D59**, 903-909
- Diederichs, K., Junk, M. (2009) Post-processing intensity measurements at favourable dose values.*J. Appl. Crys*t. **42**, 48-57
- Diederichs K. (2009) Simulation of X-ray frames from macromolecular crystals using a ray-tracing approach. *Acta Cryst.* **D65**, 535-42
- Diederichs K. (2010) Quantifying instrument errors in macromolecular X-ray datasets. *Acta Cryst.* **D66**, 733-740
- Diederichs K., "Crystallographic data and model quality" in *Nucleic Acids Crystallography* (Ed. Ennifar), Methods in Molecular Biology (Springer 2015)
- Karplus P.A. and Diederichs K. (2015) Assessing and maximizing data quality in macromolecular crystallography. *Curr.Op.Str.Biol.* **34**, 60-68
- Assmann, G., Brehm, W. and Diederichs, K. (2016) Identification of rogue datasets in serial crystallography (2016) *J. Appl. Cryst.* **49**, 1021-1028.

Thank you!

(obtain PDF from kay.diederichs@uni-konstanz.de)