

Introduction to Diffraction Data Processing using XDS
2018 Cold Spring Harbor X-Ray Methods in Structural Biology Course
Matthew J. Whitley
University of Pittsburgh

- We will use the following programs and scripts, which you will need to download and set up on Mac or Linux computers in your home lab in order to use XDS in the future:

Main XDS package: <http://xds.mpimf-heidelberg.mpg.de/>

XDSGUI: <https://strucbio.biologie.uni-konstanz.de/xdswiki/index.php/XDSGUI>

generate_XDS.INP:

https://strucbio.biologie.uni-konstanz.de/xdswiki/index.php/Generate_XDS.INP

XDS-Viewer: <http://xds-viewer.sourceforge.net/>

XDSSTAT: <https://strucbio.biologie.uni-konstanz.de/xdswiki/index.php/Xdsstat>

XDSCC12: <https://strucbio.biologie.uni-konstanz.de/xdswiki/index.php/XDSCC12>

Neggia library for handling images from Eiger detectors in .h5 format (if necessary):

<https://www.dectris.com/support/downloads/software/neggia>

- In this tutorial, you will be given several different sets of diffraction data to process using XDS. Each data set was chosen to highlight key steps and/or potential pitfalls in data processing. We will walk through the processing of the first data set together in order to gain familiarity with XDS and its tools. After we finish processing the first data set together, students will work through the additional tutorial data sets at their own pace, and I will be present to assist and answer any questions. Finally, students may also attempt to process their own data from home or from the course's data collection trip to NSLS-II during the latter part of this session or during any of the other free work periods. I will be happy to assist as needed.
-

General Tips and Advice:

- If you ask 10 different crystallographers to show you how to process data using the same processing package (such as XDS), you will almost certainly be shown 10 different workflows that all differ modestly from each other. This is not a problem, as long as each major step is accomplished appropriately. As long as the final processing results are reasonable, do not be alarmed if one person does things slightly differently than another person.
- It is possible to use XDS exclusively via XDSGUI (the XDS graphical user interface), but some things are simply easier and/or quicker to accomplish using the command line. Being (or getting) familiar with a Linux/Mac text editor such as vi, emacs, or gedit is advantageous, not just for diffraction data processing, but for your whole research career.
- It is simply not possible to teach every nuance of processing data with XDS in just a few hours. Experience is the best teacher, so the smart way to gain skill with XDS is to process as

many data sets as you can get your hands on. Process all your data from this course, regardless of whether you go on to solve and refine the structure. Reprocess existing data sets from your home lab. Download data sets from online repositories and process them. The more data sets you process, the more likely you are to come across unusual or difficult cases that force you to learn something new. *Luckily, XDS and its associated tools have detailed documentation, both online and in the literature.* Don't hesitate to search the documentation and literature if you don't understand something and want or need to learn more.

- Here are a few critical resources for information about XDS data processing. There are many more good sources than can be listed here, but these will help you get started:

General XDS documentation: http://xds.mpimf-heidelberg.mpg.de/html_doc/XDS.html

What intermediate files are used by each processing step, and what files does the step generate? http://xds.mpimf-heidelberg.mpg.de/html_doc/xds_files.html

XDSWiki: a wide-ranging resource of XDS information, tips, and case studies. It is useful to explore the wiki in detail – you'll always learn something new:

<https://strucbio.biologie.uni-konstanz.de/xdswiki/index.php/Xds>

Great review article for considering sources of error and optimizing XDS data processing:

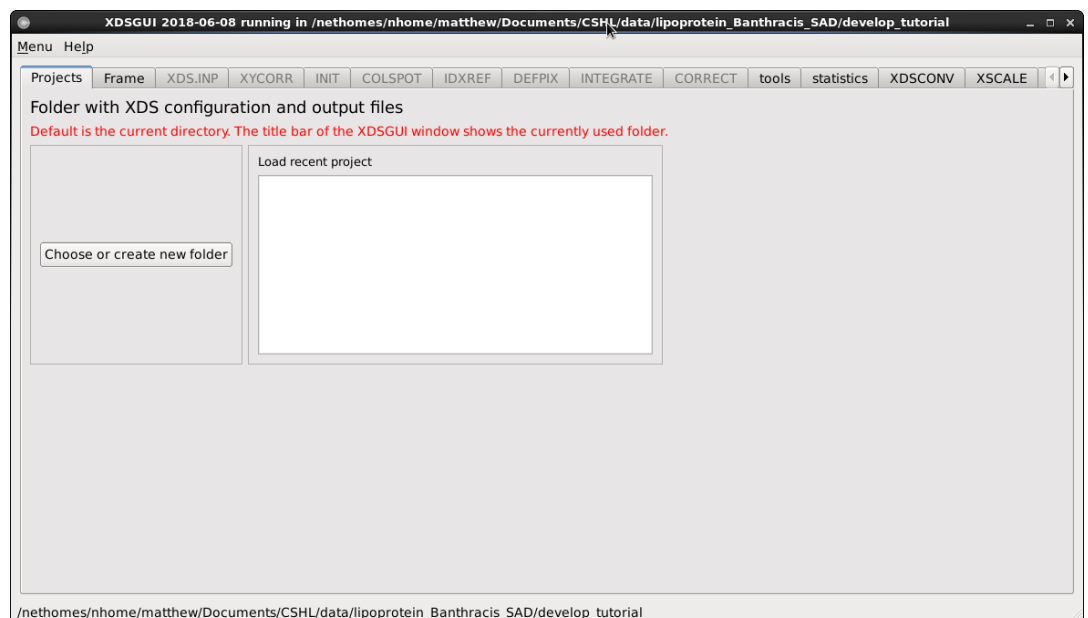
Diederichs, K. Crystallographic Data and Model Quality. *Methods Mol Biol.* (2016) 1320:147-173.

Tutorial Data Set #1: Putative lipoprotein from *Bacillus anthracis*

Data archived at proteindiffraction.org under ID 4NOHIDP05511

I recommend that you create a specific directory for processing within the general folder for this project in order to store all the processing output. Go there, and then launch XDSGUI.

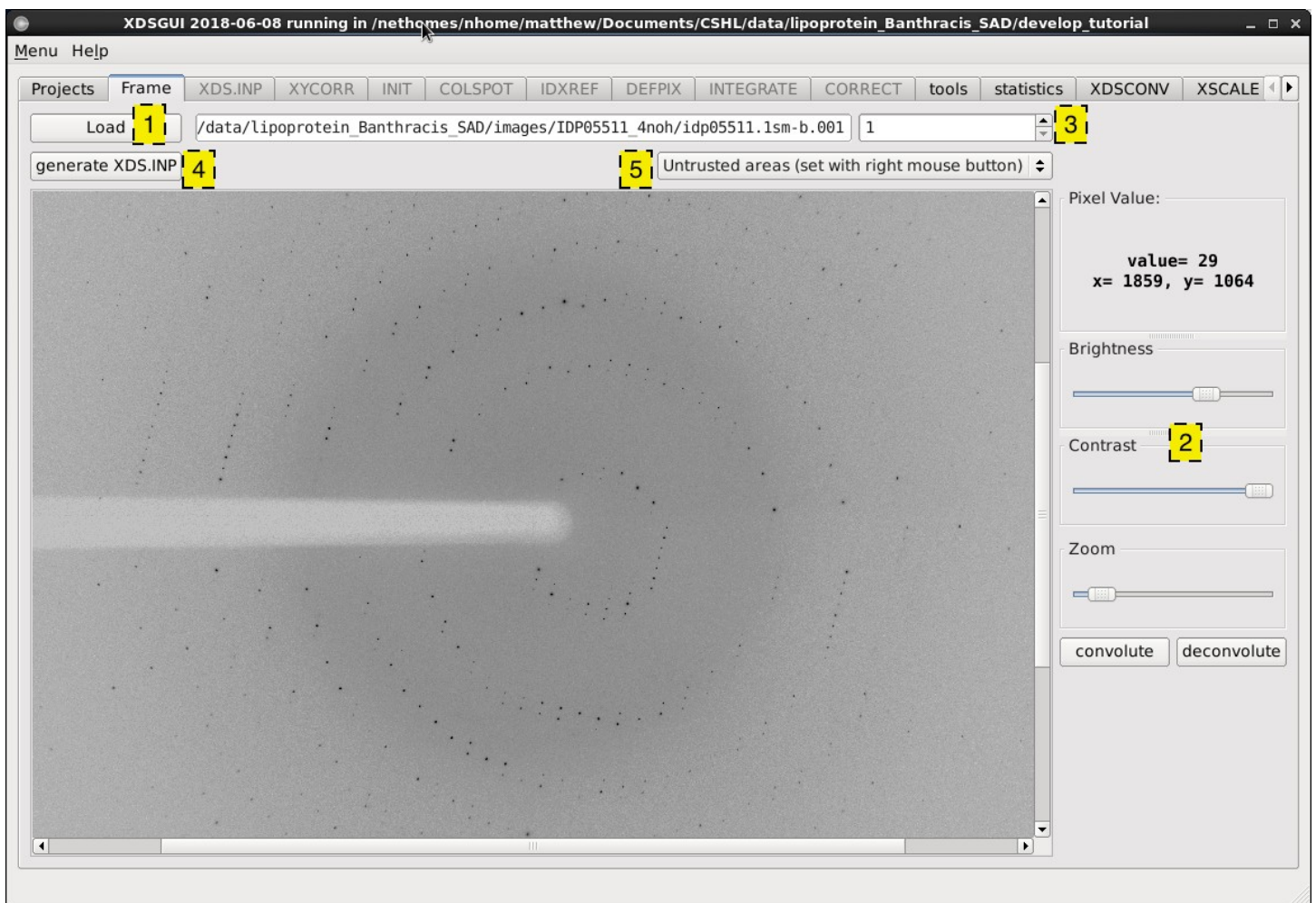
```
> mkdir process-xds
> cd process-xds
> xdsgui
```



Processing using XDS is accomplished in a series of steps called XYCORR, INIT, COLSPOT, IDXREF, DEFPIX, INTEGRATE, and CORRECT. Each step generally takes as its input the output of a previous step, and the step's output will generally be the input for a subsequent step. As each step is completed, its log file (ending in .LP) will be displayed in the corresponding tab in XDSGUI. Accessible tabs are in black; tabs currently unavailable (because the particular step has likely not yet been performed) are grayed out.

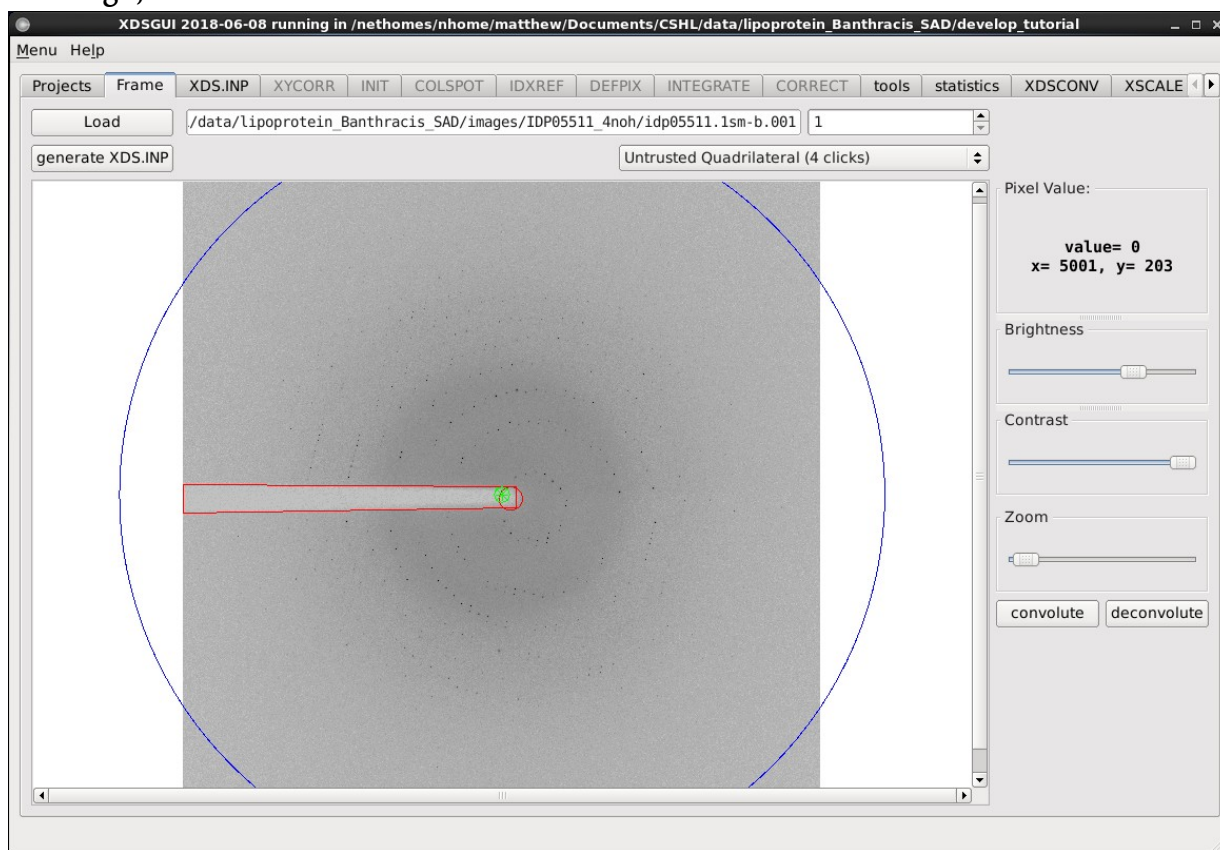
- Click on the FRAME tab. In this tab, we will load in the first diffraction image, view images in the data set, generate the input file necessary to process the whole data set with XDS, and then create beam masks.

In the FRAME tab, click Load and navigate to where your images are stored. It is a good idea to keep your images in their own separate subdirectory where they will not become interspersed with files from processing, phasing, and/or refinement. I have provided the images in a separate directory called images. XDS can handle images from most common x-ray detectors, either uncompressed or compressed in gzip or bzip2 format. Processing compressed images is slower than processing uncompressed images, but compressed images take up substantially less space on your computer's hard drive.



- (1) Click here to navigate to and load the first image, which is already done in the figure.
- (2) You can adjust how the images are displayed in order to best see the reflections.

- (3) You can scroll through the images here (or type in a particular image number) in order to visually assess the data set's quality. The scrolling process is not speedy, but it works.
- (4) **The critical step:** click generate_XDS.INP in order to create the input file (XDS.INP) required by XDS to process the data. If you forget this step, or if the step fails, you will be able to go no further and might have to manually create the input file. If the input file is generated, a green circle that signifies the direct beam position as extracted from the image headers will be displayed on the image, as well as a large blue circle that represents the outer limit of the image area that the computer will analyze to locate strong spots for indexing. Finally, the XDS.INP tab will now be black instead of gray.
- (5) Use the quadrilateral and ellipse tools to mask the beam stop and any other unreliable areas of the detector. To use the quadrilateral tool, simply right click on the four corners of a quadrilaterally shaped region that you want to exclude. To use the ellipse tool, imagine that the circular or elliptical region you want to exclude is inscribed in a square or rectangle, and then right click on the upper left and lower right corners of that square/rectangle. There is no limit to the number of excluded areas that you can create. It often helps to zoom in in order to more accurately select regions to exclude. Areas to be excluded are outlined in red on the displayed image, as below:



- Now move to the XDS.INP tab. This is the main input file XDS needs in order to process data. There are many, many parameters defined in this file, most of which were set based on information extracted from the image headers. In many cases, the XDS.INP file will be correct, but this is *not* a certainty, and **it is up to you, the human crystallographer, to verify that the essential information is correct!** Incorrect information in the input file is likely to lead to (i) an inability to process the data or (ii) substantially incorrect results!

```
! written by generate_XDS.INP version 0.83 (25-Jun-2018)
JOB= XYCORR INIT COLSPOT IDXREF DEFPIX INTEGRATE CORRECT
ORGX= 2048.00 ORGY= 2048.01 ! values from frame header; only read by XYCORR, IDXREF

DETECTOR_DISTANCE= 180.000 !read by XYCORR, IDXREF. Negative if detector normal points to crystal.
OSCILLATION_RANGE= .750
STARTING_ANGLE= 80.00
X-RAY_WAVELENGTH= .97911
NAME_TEMPLATE_OF_DATA_FRAMES=/nethomes/nhome/matthew/Documents/CSHL/data/lipoprotein_Banthracis_SAD/images/IDP05511_4noh/idp05511.1sm-b.???
! REFERENCE_DATA_SET=xxx/XDS_ASCII.HKL ! e.g. to ensure consistent indexing
DATA_RANGE=1 300
SPOT_RANGE=1 150
! BACKGROUND_RANGE=1 10 ! rather use defaults (first 5 degree of rotation)

SPACE_GROUP_NUMBER=0 ! 0 if unknown
UNIT_CELL_CONSTANTS= 70 80 90 90 90 90 ! put correct values if known
INCLUDE_RESOLUTION_RANGE=50 0 ! after CORRECT, insert high resol limit; re-run CORRECT
! IDXREF now obeys INCLUDE_RESOLUTION_RANGE and EXCLUDE_RESOLUTION_RANGE to exclude ice-rings

FRIEDEL'S_LAW=FALSE ! This acts only on the CORRECT step
! If the anom signal turns out to be, or is known to be, very low or absent,
! use FRIEDEL'S_LAW=TRUE instead (or comment out the line); re-run CORRECT
```

Most of the critical parameters to check are toward the top of the file.

-The JOB line tells XDS which steps to perform when run. An exclamation point (!) signals a comment; XDS will not read anything to the right of an exclamation point. So, for instance, if you want XDS to stop after indexing, simply put an exclamation point after IDXREF, and the steps DEFPIX, INTEGRATE, and CORRECT will not be performed. Blank lines are also ignored.

-ORGX and ORGY are the detector coordinates of the beam center. These coordinates are taken from the image header, *but it is not uncommon for them to be incorrect. Incorrect beam center coordinates are a frequent source of indexing failure!* The best way to avoid this problem is to actually measure the beam center coordinates before data collection, for example by recording either a phosphor image or a highly attenuated direct beam shot. If these measurements are not available, there are other ways to estimate the beam coordinates, but actually measuring the coordinates is the safest thing to do.

-DETECTOR_DISTANCE (in mm), OSCILLATION_RANGE (in degrees) and X-RAY_WAVELENGTH (in Å) are self-explanatory, but it is critical to make sure these parameters are accurate.

-NAME_TEMPLATE_OF_DATA_FRAMES tells the name of your images and the path to where they are stored. XDS assumes that all images in your data set have the same base name and an incremented image number (for example image_0001.cbf, image_0002.cbf ...). DATA_RANGE gives the number of the first and last image in your data set. In this tutorial, there are three hundred images numbered 1-300. If, for some reason, the first image you want to process is numbered 7, and the last image is 241, then you would set this to DATA_RANGE=7 241. DATA_RANGE must be set to a continuous range of images; if you want to exclude individual images within the overall range, you can add an EXCLUDE_DATA_RANGE parameter.

-SPOT_RANGE gives the range of images that XDS will search for strong reflections to use for indexing. By default, an XDS.INP input script generated as we have done here will use the first half of the images, no matter how many total images there are. This is often perfectly

satisfactory, *but if indexing fails you may consider altering this parameter* to use a smaller number of images or to distribute the images used throughout the whole data set. You may enter multiple SPOT_RANGE lines, but each must be set to a continuous range of images.

-SPACE_GROUP_NUMBER and UNIT_CELL_CONSTANTS are self-explanatory. The cell parameters are given to UNIT_CELL_CONSTANTS in this order: a b c alpha beta gamma

If SPACE_GROUP_NUMBER is set to zero (unknown), then XDS will ignore UNIT_CELL_CONSTANTS completely. That is, if the space group is entered as unknown, the cell constants are also assumed to be unknown

-INCLUDE_RESOLUTION_RANGE allows you to set the low and high resolution limits (in that order!) to be used for processing and calculation of statistics. If the high resolution limit is set to zero, the entire image will be processed.

-FRIEDEL' S_LAW determines whether the reflections (h k l) and (-h -k -l) are expected to have the same intensity. TRUE = yes FALSE = no

Every parameter in the input file is explained in the XDS documentation, but the parameters given above are the most critical. Consult the documentation for further details.

If you have made any changes to the input file, *be sure to click Save*. Finally, click Run XDS. XDS will now perform all the steps listed in the JOB line of the input file. The unavailable gray tabs will become black and available when the given steps are complete. Also, as XDS runs, information will scroll by in the terminal window that was used to launch XDSGUI.

• After several minutes, you will notice that XDS has run to completion without giving any error messages. Theoretically, we are now completely finished with our data processing, and you are ready to proceed to the next step of your workflow, such as phasing.

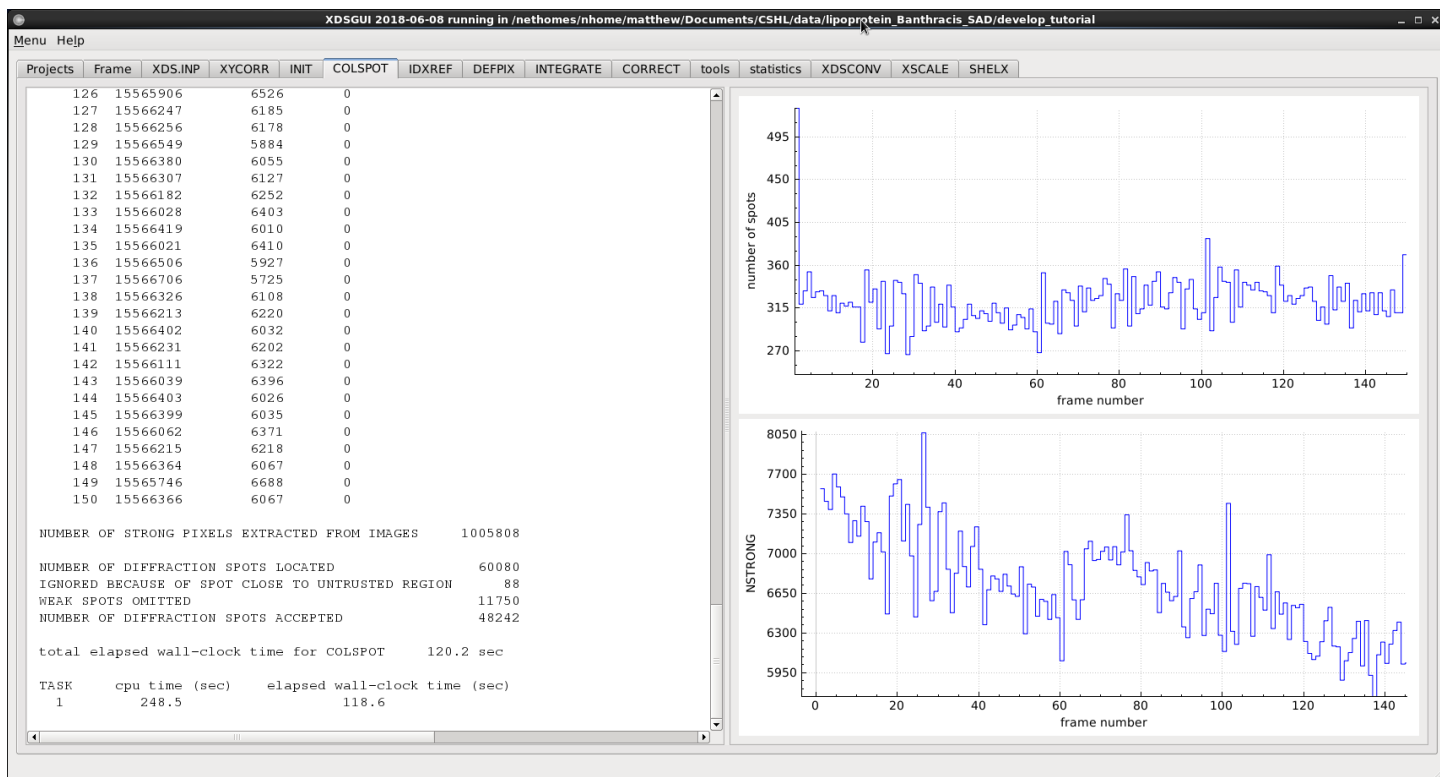


**PAY
ATTENTION**

In practice, however, it is essential to remember that data processing programs are not infallible. They are only as good as the data you feed them, and for better or worse, they do exactly what you tell them to do and nothing more (or less). It is **critical** that you now inspect the processing results to make sure they are logical and reasonable. It is also usually possible to optimize and repeat the processing cycle and, ultimately, to achieve improved results.

• We will now focus on key portions of the XDS output to help us determine whether our processing results are reasonable.

One thing to look at is the results of spot finding (the COLSPOT step), which used the images you gave in the SPOT_RANGE parameter in XDS.INP. Click on the COLSPOT tab. The COLSPOT.LP log file is displayed on the left, with graphical output on the right.

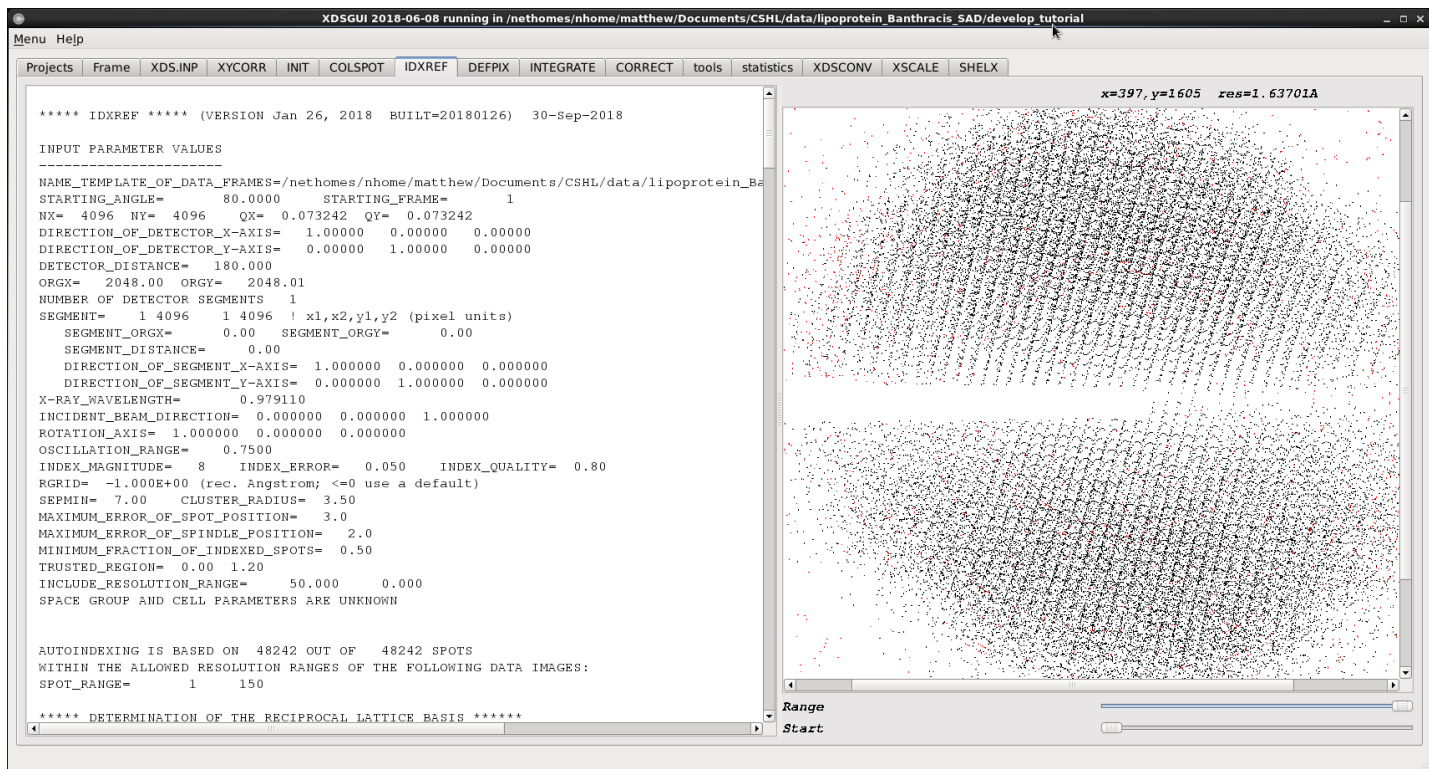


The text output shows the number of strong pixels found for each image examined. The right side shows the number of strong reflections and the number of strong pixels (top and bottom, respectively) as a function of image number. If indexing succeeded, then everything was probably fine with the COLSPOT step, but if you see abrupt drops (especially to near zero) for any images, or especially for a range of images, then it could indicate some problem such as the crystal rotating out of the beam. In that case, you might want to process again with those images excluded. In this particular case, these plots appear perfectly acceptable.

More critically, we now want to determine whether the results of indexing appear to be reliable. Click on the tab **IDXREF**. The text of the output file **IDXREF.LP** is again on the left, and graphical information on the right.

The image on the right side is of the spots that were found by COLSPOT. Reflections that were successfully indexed are colored black, whereas reflections that were not indexed are colored red. In general, a good indexing result yields a large number of black reflections and a high ratio of black to red reflections (many indexed reflections, few unindexed reflections). If there are a large number of black reflections but also a large number of red reflections, it is possible that your images contain diffraction from multiple lattices and that XDS indexed one of them. If this is the case, it is possible to return later to attempt to index and process other lattices. If there are a large number of red reflections but a small number of black reflections, you should be cautious about the reliability of the indexing results.

Now look at the text information on the left side. We will examine the numbers to convince ourselves that our positive general impression from the graphical information on the right is valid.



How many spots were found as input for indexing?

AUTOINDEXING IS BASED ON 48242 OUT OF 48242 SPOTS
WITHIN THE ALLOWED RESOLUTION RANGES OF THE FOLLOWING DATA IMAGES:
SPOT_RANGE= 1 150

This shows that 48242 spots were found by COLSPOT. As long as this isn't a very small number (which we already verified by looking at the COLSPOT tab, anyway), things are probably fine. Indexing can theoretically proceed from a modest number of spots from a single image, but by using a larger number of spots from many images, we're tacitly allowing indexing to take into account small frame-to-frame variation that might have occurred over the data collection. If the frame-to-frame variations are too large, however, indexing might fail completely.

Next, go further down the output file to look at:

CLUSTER COORDINATES AND INDICES WITH RESPECT TO REC. LATTICE BASIS VECTORS

#	COORDINATES OF VECTOR CLUSTER			FREQUENCY	CLUSTER INDICES		
1	0.0033651	-0.0020003	0.0128540	2624.	1.00	0.00	-0.00
2	-0.0235526	0.0010498	0.0063176	2527.	-0.00	-1.00	-0.00
3	-0.0269174	0.0030517	-0.0065317	2506.	-1.00	-1.00	0.00
4	0.0010180	-0.0227716	0.0102465	2447.	1.00	0.00	-1.00
5	0.0023478	0.0207683	0.0026012	2400.	-0.00	0.00	1.00
6	-0.0201877	-0.0009488	0.0191761	2392.	1.00	-1.00	-0.00
7	-0.0067295	0.0039992	-0.0257133	2386.	-2.00	-0.00	0.00
8	0.0258992	0.0197210	-0.0037159	2358.	-0.00	1.00	1.00
9	0.0302817	-0.0050523	0.0193905	2332.	2.00	1.00	-0.00

10	-0.0225346-0.0217199 0.0165697	2321.	1.00	-1.00	-1.00
...					
55	-0.0134565 0.0079948-0.0514348	1922.	-4.00	0.00	-0.00
56	-0.0144752 0.0178269 0.0346408	1899.	2.00	-1.00	1.00
57	-0.0289545 0.0485912-0.0270326	1892.	-3.00	-1.00	2.00
58	-0.0181522-0.0464844 0.0396758	1890.	3.00	-1.00	-2.00
59	-0.0359935-0.0137247-0.0348513	1887.	-3.00	-1.00	-1.00
60	0.0517958 0.0394417-0.0074267	1884.	0.00	2.00	2.00

Take a look at the cluster indices in the last 3 columns. **If these are not almost exactly integers, then something has certainly gone wrong, and the output unit cell parameters are not reliable.** For example, in the first line the cluster indices are the integers 1.00 0.00 -0.00. A near integer output such as 1.02 0.01 and -0.01, for example, would be ok, but an output of 1.21 0.11 and -0.15 would indicate an error (even a non-integer output in only one of the three columns indicates problems). In our current case, all the cluster indices are integers, and thus we are gaining confidence that the indexing results are valid.

Next we look for quantitative information about the number of indexed vs. unindexed reflections.

```

***** INDEXING OF OBSERVED SPOTS IN SPACE GROUP # 1 *****
45012 OUT OF 48242 SPOTS INDEXED.
4 REJECTED REFLECTIONS (REASON: OVERLAP)
3226 REJECTED REFLECTIONS (REASON: TOO FAR FROM IDEAL POSITION)
EXPECTED ERROR IN SPINDLE POSITION 0.125 DEGREES
EXPECTED ERROR IN DETECTOR POSITION 0.94 PIXELS

```

In agreement with the fact that we saw a large number of black spots and a small number of red spots in the IDXREF tab, we see that we were able to index more than 45,000 out of around 48,000 spots. It is a good sign that a single indexing solution was able to account for such an overwhelming percentage of the detected reflections.

Finally, we get information about which Bravais lattices our indexed reflections are compatible with and the parameters of the lattices.

A lattice character is marked "*" to indicate a lattice consistent with the observed locations of the diffraction spots. These marked lattices must have low values for the QUALITY OF FIT and their implicated UNIT CELL CONSTANTS should not violate the ideal values by more than

```

MAXIMUM_ALLOWED_CELL_AXIS_RELATIVE_ERROR= 0.03
MAXIMUM_ALLOWED_CELL_ANGLE_ERROR= 2.0 (Degrees)

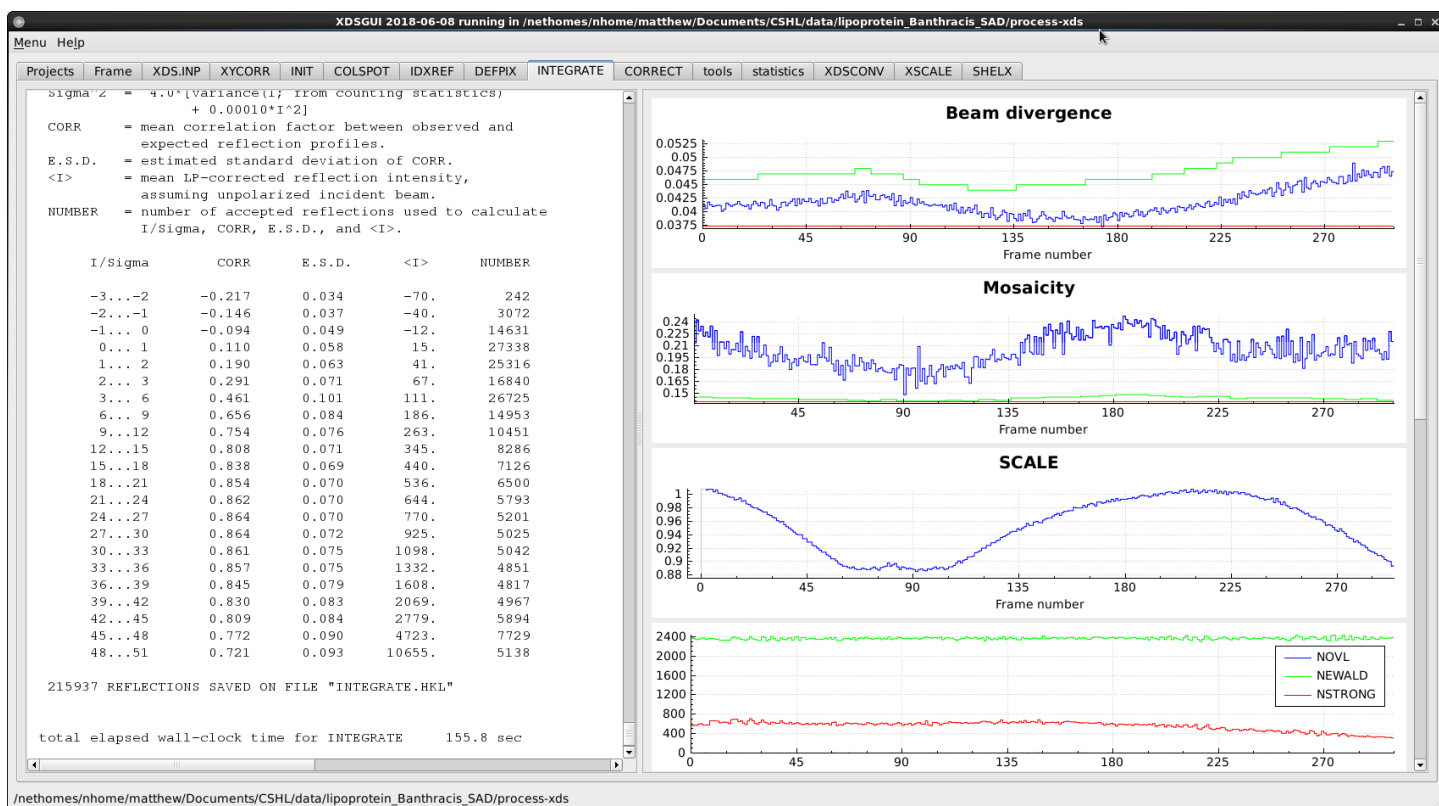
```

LATTICE- CHARACTER	BRAVAIS- LATTICE	QUALITY OF FIT	UNIT CELL CONSTANTS (ANGSTROMS & DEGREES)						
			a	b	c	alpha	beta	gamma	
* 44	aP	0.0	41.0	47.5	74.3	90.1	90.0	91.9	
* 31	aP	0.5	41.0	47.5	74.3	89.9	90.0	88.1	

*	34	mP	1.9	41.0	74.3	47.5	90.1	91.9	90.0
*	35	mP	19.7	47.5	41.0	74.3	90.0	90.1	91.9
*	33	mP	20.6	41.0	47.5	74.3	90.1	90.0	91.9
*	32	oP	21.1	41.0	47.5	74.3	90.1	90.0	91.9
	14	mC	172.8	61.7	63.7	74.3	90.0	90.1	81.6
	13	oC	173.7	61.7	63.7	74.3	90.0	90.1	81.6
	10	mC	174.2	63.7	61.7	74.3	89.9	90.0	98.4
	11	tP	192.9	41.0	47.5	74.3	90.1	90.0	91.9

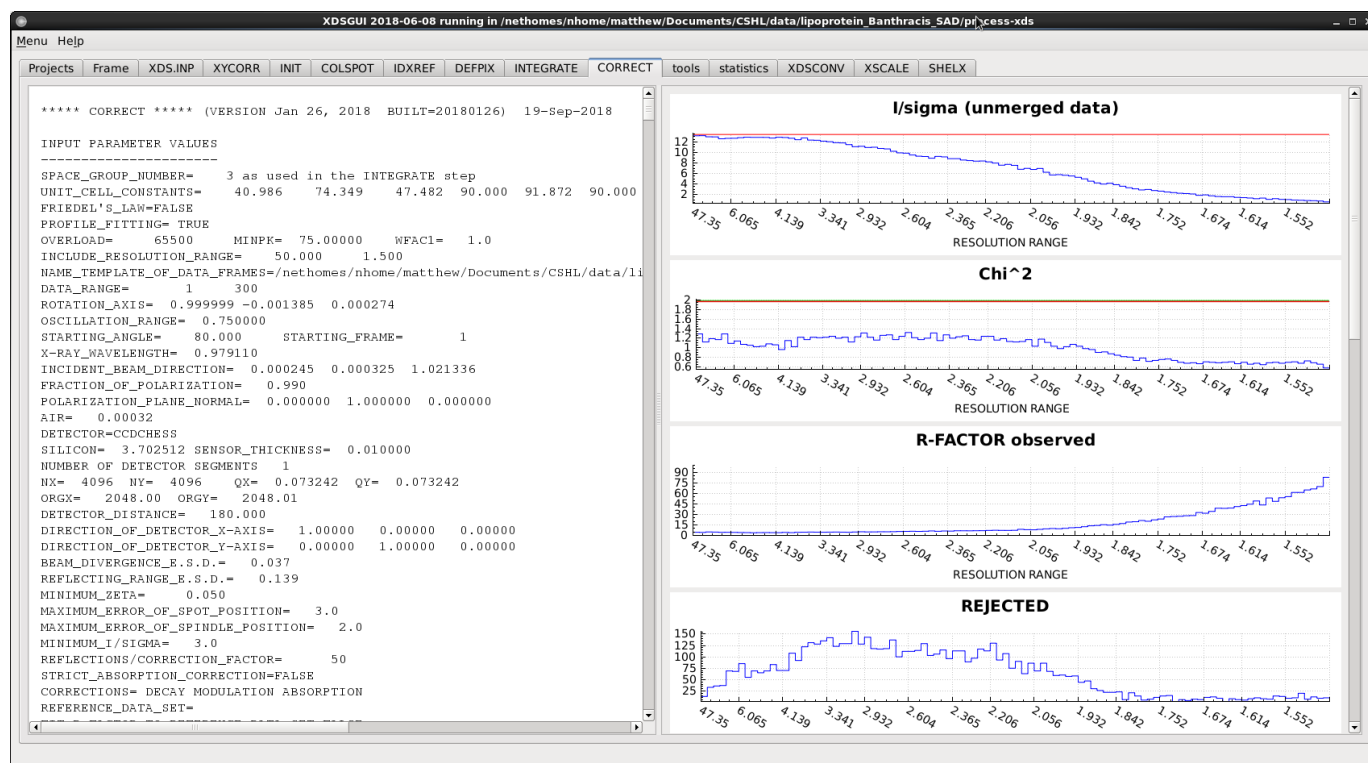
Thus, our indexed data are generally compatible with primitive triclinic (aP), primitive monoclinic (mP), and primitive orthorhombic (oP) lattices. All the other lattice types have a bad Quality of Fit (high numbers). Which one of the possible solutions in particular is likely to be correct? A primitive triclinic solution is always possible and will always have the best Quality of Fit, so we want to determine whether any of the possibilities with higher symmetry (monoclinic or orthorhombic) could be correct. Looking at the progression of Quality of Fit numbers for the possible lattices, we see that the first mP solution has a barely worse Quality of Fit (1.9 vs 0.0/0.5) than the triclinic solutions. This suggests, *but does not prove*, that the lattice might be at least primitive monoclinic. We then note that the jump in Quality of Fit after the first primitive monoclinic solution is much more substantial, indicating that these higher symmetry lattices are noticeably worse fits to our data. At this point, it seems reasonable to assume that we have a primitive monoclinic Bravais lattice and unit cell constants of 41.0 74.3 47.5 90.1 91.9 90.0. Notice that, at this step, geometric requirements for a monoclinic unit cell such as $\alpha = 90$ have not yet been imposed. The values given here are rough unit cell measurements. How can we gain further confidence that the higher symmetry solutions are not correct? At the end of processing we can try to merge the data assuming the higher symmetry; if the merging statistics are poor, the higher symmetry solutions are likely incorrect.

- Next click the INTEGRATE tab. As usual, the contents of the log file INTEGRATE.LP are displayed on the left, with graphical output on the right. The log file contains detailed information about profile fitting, but in many cases you do not need to manually inspect this information. The graphical output on the right is more useful. The values of numerous parameters as a function of frame number are graphed here. **As a general rule of thumb, wild oscillations or spikes in the graphs are a bad sign, as are drastic changes in a parameter's value.** Look, for instance, at the scale graph. We notice that it varies relatively smoothly over the data set and that the absolute variation is between 0.9 and 1.0; there are no sharp or wild variations, and the magnitude of the variation is small. This is a reasonable result. A large increase in the scale factor with increasing frame number could, for example, indicate that substantial radiation damage has occurred, and the scale factor is attempting to compensate for the damage. As a result, you might consider reprocessing the data with the latter, damaged frames excluded.



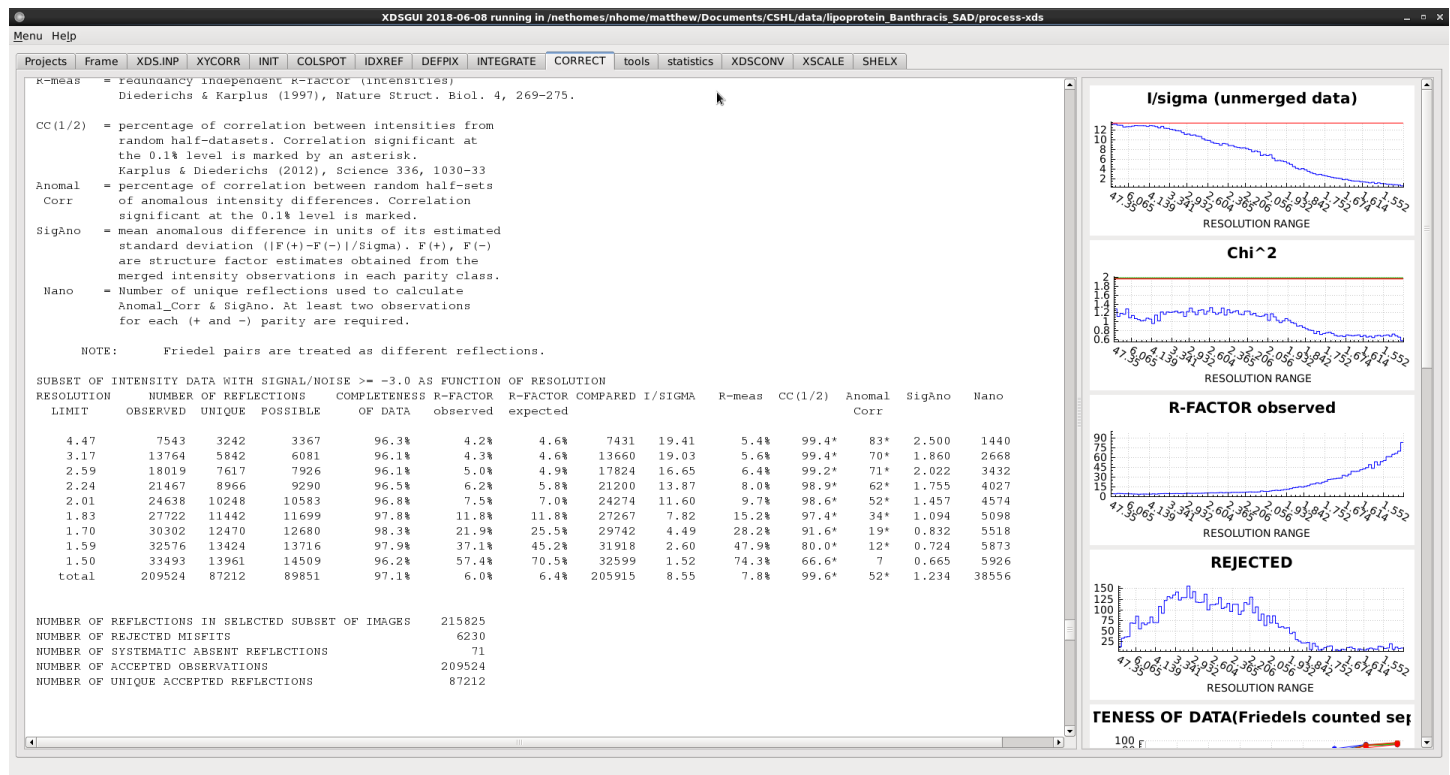
Note: the integrated reflections are stored in the output file INTEGRATE.HKL. *These reflections are unscaled and unmerged!*

• Now click on the CORRECT tab. The CORRECT step applies various empirical corrections to the intensities measured during integration, refines the unit cell constants, and scales the data.



As usual, the contents of the log file CORRECT.LP are shown on the left, with graphical output on the right. The graphical information contains much useful information about the quality of the final processed data. Inspecting it can help you detect ice rings, problematic frames, as well as give you some idea of the practical resolution limit of your data.

After all processing steps are complete, most people want to know what “the statistics” for the data set are. These are most easily found in the CORRECT.LP output file (left window) by scrolling *all the way to the bottom*, and then slowly scrolling back up until the first data table is found that looks as follows:



Be careful – the CORRECT.LP output file is filled with numerous such tables, but all tables except the very last one refer to intermediate stages of processing; *only the last table contains the statistics for the entire processed data set*. Here you will find all the typical information such as completeness, R_{merge} , R_{meas} , $I/\text{sig}(I)$, $CC_{1/2}$, and so forth. Note that XDS does not explicitly print the average multiplicity, but this can be calculated easily from the information given here as “Number of Accepted Observations” divided by “Number of Unique Accepted Observations.”

We can perform additional analysis of our data by moving to the statistics tab. If you have downloaded and placed xdscc12 and xdsstat in your path, clicking on the buttons run xdscc12 and run xdsstat will perform these analyses and display the results graphically in the window. These diagnostics can help you identify problematic data frames and the accumulation of radiation damage, among other things.

Note: the CORRECT step outputs the final *scaled but unmerged* reflections to the file XDS_ASCII.HKL. This file can be used directly in many phasing and refinement program, or you can convert it to .mtz format for use with other programs.

What to do now?

Now that we have performed one complete round of data processing, we will update our input parameters and repeat the processing in an attempt to optimize and improve the results.

Updates to consider:

1. After the first round of processing, update SPACE_GROUP_NUMBER in XDS.INP with the space group currently chosen for scaling, which is found by viewing XDS_ASCII.HKL in a text editor.
2. Update UNIT_CELL_CONSTANTS in XDS.INP with the refined values, which can be found by viewing the file XDS_ASCII.HKL in a text editor.
3. Add the refined beam divergence and reflecting range to XDS.INP so that further refinement will start from these updated values. Click the tools tab, then Optimizing data quality, then Copy BEAM_DIVERGENCE, REFLECTING_RANGE from INTEGRATE.HKL to XDS.INP. Click 'reload' when XDSGUI tells you that XDS.INP has changed.
4. Update refined geometric parameters of the experiment. Click the tools tab, then Optimizing data quality, then Copy latest geometry description over previous one. This will replace the existing file XPARM.XDS with the refined values contained in GXPARM.XDS.
5. Update the high-resolution limit of INCLUDE_RESOLUTION_RANGE in XDS.INP to a suitable value based on your analysis of the CORRECT output.

Then update the JOB line in XDS.INP to re-run the INTEGRATE and CORRECT steps. You can iterate this process until the unit cell constants, beam divergence, reflecting range, etc., all cease changing substantially from iteration to iteration. *There are also other parameters that can be updated* to possibly improve the data processing results; consult the online XDS documentation for additional details.

Finally, you will need to settle on what the true space group might be. In the indexing stage, we determined likely Bravais lattices compatible with the observed data. However, there are many space groups possible for a given lattice. To determine likely space groups, we first need to determine the Laue class (the symmetry of the diffraction pattern, the point group of the crystal plus a center of symmetry due to Friedel's Law) and then check for the presence of translational symmetry components such as screw axes that manifest as systematic absences. The program pointless, part of the CCP4 package, can assist with this process. In XDSGUI, click the tools tab, then further analyses, then determine space group with pointless. The output will be contained in pointless.log in the directory where XDS is running. In the present case, pointless detects the presence of a 2_1 screw axis, and thus the primitive monoclinic space group $P2_1$ is a likely candidate instead of just $P2$. Update the XDS.INP file with this new information. You can re-integrate if desired, but only the CORRECT step is required to be re-run in order to scale the data in the new space group. Finally, remember that pointless *is not infallible!* The reliability of its suggestions is heavily dependent on things like general data quality and quality of data processing. Check the merging statistics in the space group suggested by pointless to make sure that they are reasonable! Furthermore, certain space groups (enantiomorphic space groups, such as $P4_12_12$ and $P4_32_12$) cannot be distinguished on the basis of systematic absences!

Whenever you are finally satisfied with the processing results and are ready to move on to another step of structure solution, such as phasing, you could merge symmetry-equivalent reflections, add Rfree flags, and convert the data to .mtz format using XDSCONV (or you could use the scaled but unmerged reflections file XDS_ASCII.HKL directly in programs capable of reading it such as Phenix and shelx). See the XDS documentation for information about how to use XDSCONV.

Additional practice with this data set, when time permits:

- Does your analysis suggest the presence of significant anomalous signal? If so, try to obtain phases via an appropriate experimental phasing technique.
- Obtain phases via molecular replacement.
- However you phase the data, build the model and refine the structure.
- Reprocess this same data set using a different data reduction program such as DIALS or Mosflm and compare the results to those obtained using XDS.

Tutorial Data Set #2: BOA (*Burkholderia oklahomensis* agglutinin), an antiviral lectin

Data provided by M. Whitley

This data set contains 180 images collected several years ago at the Advanced Photon Source from a selenomethionine-labeled BOA crystal. Process this data set with XDS using the same workflow introduced in Tutorial Data Set #1. Review the outputs of spot finding, indexing, integration, and scaling. Does anything give you pause? If something has gone wrong, diagnose and correct the problem so that the data set can be processed successfully. After you have processed the data appropriately, you might eventually want to use the data to practice Se-SAD phasing, model building, and model refinement.

Tutorial Data Set #3: Mouse Sidekick-2 Immunoglobulin Domains 1-4, H18R/N22S Mutant

Data obtained from the SBGrid Data Bank, ID# 373

This data set comprises 180 images collected several years ago at Sector 24 of the Advanced Photon Source. This data set has more than one issue that should capture your attention during processing. Process the data using XDS and analyze the output of each step to discover the troublesome issues. Determine and implement a course of action that will lead to successful processing of the data.

Tutorial Data Set #4: iMosflm Tutorial Example 1

Data obtained from Harry Powell, MRC Laboratory of Molecular Biology

There are only five images in this data set. Your goal is to index the data correctly, but several complications make this not straightforward to accomplish. However, the complications are fairly common in real data, so figuring out how to diagnose and overcome them is likely to be useful to you in your own work at some point. Hint: carefully examine the agreement between the observed and predicted reflection locations as a function of image number.