Tutorial for Processing BioCARS Laue Serial Crystallography Data using Precognition/Epinorm and in-house Phyton program pyPrecognition

Prepared by: Vukica Srajer, June 2025

- Precognition/Epinorm is a licensed software for processing of Laue crystallographic data (Renz Research, Inc). To obtain this software package, please contact Zhong Ren at zren@uic.edu. Tutorials and the manual for using the package are provided on: https://biocars.uchicago.edu/facilities/software/precognition-documentation/

- Precognition/Epinorm was originally written for processing of single-crystal Laue data. For processing of serial crystallography data, efficient processing of images had to be done in parallel so Robert Henning at BioCARS developed Phyton GUI, pyPrecognition.py, for such processing.

- Here we provide a tutorial on using pyPrecognition.py.

This tutorial is based on a BioCARS Laue serial crystallography data set. If you want to use this example data set to practice data processing, please contact Robert Henning (henning@cars.uchicago.edu).

- Data set is generously provided by Seong Ok Kim, Department of Chemistry, KAIST, Center for Advanced Reactions Dynamics, Institute of Basic Science, Republic of Korea
- Serial crystallography Laue data collected in February 2022
- Sample: lysozyme
- Input parameters and other information for the example data set

    - Input cell parameters, room T: 78.4 78.4 37.9 90 90 90, P43212, space group number 96
    - Input crystal-to-detector distance: 362.5 mm
    - Input beam center: 1987.3, 1974.9
    - Number of images collected: 13,432
    - Hits selected for processing: 990
        - Selection done by BioCARS pyPrecognition.py hit-finding script
        - Selection criteria: more than 100 diffraction spots; minimum intensity above the background to be considered as a diffraction spot: 50

==========

Initial setup of pyPrecognition.py

==========

To run pyPrecognition.py: type pyPrecognition.py in data processing directory

- Set up versions of Precognition under File/Precognition Versions. Set up “precognition” as the primary version (this is version T5.2.2). Set up “precognition_old” as the secondary version.
- For the indexing step, one can select indexing with both versions in the Index tab. This will make two indexing passes: first one with the primary version, second with the secondary version for files that did not index with the primary version.
- Two versions do not always index successfully same images, with “precognition_old” typically indexing successfully more images than “precognition”.
- Once some parameters are entered, pyPrecog_settings.py file is created in the working directory. These parameters will be loaded if you quit pyPrecognition.py and restart it in the same directory.
- In the "Setup" tab, set up:

Detector: RayonixMX340

Center: 1987.3, 1974.9 (these numbers are for the example data set; ask beamline scientist for the best values for beam center for your experiment)
Pixel size: 0.0886, 0.0886
Diagnostics: off
Busy: off
Process directory: directory where you are processing data
CPU's: Check how many CPUs your computer has. Since indexing is a time consuming step, you can select how many CPUs to use in parallel for indexing to speed up data processing.

==========

Hit finding

==========

Use "Hitfinder" tab for hit finding. Hit-finding is typically run during the beamtime in each data directory, using the low hit-finding criteria (default is 20, 20: at least 20 diffraction spots per image, >20 intensity above the background is required to be considered as a diffraction spot) to find spots. This is done using "Process" in the "Hitfinder tab". This does take some time to complete. A hitfinder_stats.txt file is created in each data directory.

If this is the case, you don't have to run "Process" (hit-finding) again but you may still want to adjust the hit-finding criteria for the images you want to try to index. Indexing is a slow process, depending on how many CPUs you can use (see Indexing below for an example) so you may want to skip images where just noise is picked up with the low default hit-finding parameters).

- Use "Add" to add directories of interest where the data images you are processing are and see the number of hits found for hit-finding criteria listed in first two input fields (last two input fields list resolution range).
- If hit-finding criteria are too low, most images might be selected. Vary the criteria to select fewer hits. In case of the example data set:

30 20 criteria: 1406 hits
30 30 criteria: 1278 hits
100 50 criteria: 990 hits

In this case, hit-finding criteria of 100, 50 were used for indexing. This selects sufficient number of images for this strong data set (for weak data sets, one needs to lower hit-finding criteria, possibly to 30,20).

Output file:

- Use "Create list" to save the list of filenames of these hits in the working directory: images-to-index.txt.

==========

Indexing

==========

Parameter values below are for the example data set.

Make sure these are set up in the "Setup" tab (see above "Initial setup of pyPrecognition.py"):

- beam center: 1987.3 1974.9 (important to have this as accurate as possible; talk to beamline scientists about measuring this)
- pixel size: 0.0886 0.0866 (Rayonix detector pixel size)
- Set up processing directory
- CPU's: select how many CPUs you want to use in parallel for indexing images. Since indexing is time consuming step, higher number of CPUs will speed up data processing (but check how many CPUs your computer has).

"Index" tab:

- Select "Image list from file"
- “Image file” – select images-to-index.txt (990 images for the example data set)
- “Unit cell” - enter unit cell parameters (78.4 78.4 37.9 90 90 90 for the example data set)
- “Distance” – enter distance (362.5mm used for the example data set)
- Leave “Omega” and “Goniometer” default values (-90,0 and 0,0,0 respectively)
- “Resolution” – set up resolution limits for indexing (2.3, 100A for the example data set)
- “Wavelength” – wavelength range for the BioCARS X-ray spectrum (standard numbers 1.02, 1.18A, reference wavelength where relative intensity is set to 1: 1.04A)
- “Spots” - 6 4 3
  First two numbers are estimates for overall length and width of diffraction spot. Third number is a sigma level for finding spots for indexing. You may need to play with sigma level if too many or two few spots are selected (see Precognition manual). You can see selected spots after indexing. Sigma level 3 is a good starting point.
- Patterns: 0
  This is indexing option, see Precognition manual. 0 is most successful indexing option.
- Make sure to click on "Create database" - this creates database.txt file that will be updated eventually updated during geometry refinement and integration.

Click on “Index all” to index images from the images-to-index.txt list.

For the example data set:

- 489 images are reported indexed of 990 images/hits.
- It took 4.1h to index all 990 hits on a BioCARS computer with 20 CPUs used in parallel.

Assessing indexing:

- Some images reported as indexed will possibly contain multiple diffraction patterns (several crystals exposed). Some images might be mis-indexed.
- You can view images at this point - click on “View Images”. A “View images” window will open as well as an Adxv window. In “View images” window, select the first image from the drop-down list to display the image in the Adxv window. Click “Next” to display the next image.
- All images in the images-to-index.txt are shown are listed, indexed of not. Spots picked for indexing are shown in blue, predictions (for images that are indexed) are shown in green. If only blue spots are shown, image was not indexed successfully.
- For the example data set:
  - check image run5_00048 - multiple diffraction patterns recorded (multiple crystals exposed), image not indexed
  - Check image run5_00095 - image that is actually mis-indexed
  - Check image run5_00175 and run5_00393 - images that are successfully indexed.

Files used for and created after indexing:

- For each image, a number of files are created:
  - *_1_index.inp - input script (commands) for indexing based on GUI inputs
  - *_1_index.inp.log - log file from indexing
  - *_1_index.re.spt - list of found/recognized spots (coordinates)
  - *_1_index.pre.spt - list of predicted spots after the image has been indexed (hkls and coordinates)
  - *-1_index.pre.spt.inp - parameters resulting from indexing, including the orientation matrix; used as input for the next step of processing, geometry refinement

- indexing_stats.txt file is created and lists RMSD and number of matched observed/predicted spots for the RMSD for images that are reported as indexed. Click "View log" to display.
- indexing_to_refine.txt is created with a list of images reported as indexed for which geometry will be refined in the next step.

- Database.txt is updated and images that are reported as indexed are marked as such. These marked images will be used for geometry refinement.

==========

Refine geometry, calibrate (additional geometry refinement steps) & check for multiple diffraction patterns.

==========

"Refine" tab

- Set "errors" for various parameters (unit cell, distance, beam center etc) to small values (0.02 or 0.05 for beam center, cell and distance, for example). These values limit how much the parameters can change during the refinement. In some cases, when RMSD after indexing is very high and number of matched spots small, you may need to increase these errors in order to increase the parameter search space during refinement to achieve better refinement.
- At least one cell parameter is held fixed automatically in Laue refinement. In this case, a=b will be fixed so error parameters for a and b can be set to 0. Also, 90deg or 120deg cell parameters will be also held fixed.
- For resolution, wavelength, spot size and sigma level you can use the same values you used for indexing.

Click on "Refine".

Assessing refinement:

- You can again "View images". Click on View images in Refine tab. Then click on "Refine only" in the “View images” window that opens to display only indexed and refined images.
- You can select if you want to display actual spot marks (blue), prediction marks (green), both or none. Index, refine, calibrate, integrate etc check marks have to do with updates to database.txt file for marking images for the next processing steps.
- If you change integration mark for an image (if you want to include it or exclude it from integration), you need to click "Save" to save the change for that image.

Files used for and created after refinement:

- For each image, a number of files are used and created:
  - *-1_index.pre.spt.inp - parameters from the indexing step, used as input for refinement
  - *_1_refine.inp - input script (commands) for refinement created based on GUI inputs
  - *_1_refine.inp.log - log file from refinement
  - *.mccd.re.spt - list of found/recognized spots (coordinates)
  - *.mccd.pre.spt - list of predicted spots (hkls and coordinates0
  - *.mccd.inp - parameters resulting from refinement; these are used for input for the next, calibrate step; this file is updated after each calibrate step

- refinement_stats.txt - refinement log file. You can view the log files with "View log". Last two columns are RMSD and number of matched spots. RMSD should ideally be <0.5 or as low as possible and number of matched spots as high as possible.
- database.txt file has been updated with refinement information.

~~~~~~~~~~

"Calibrate" tab
~~~~~~~~~~

- Since geometry refinement typically does not converge after the first cycle, additional refinement steps are necessary. They are called "calibrate" steps since one could reset parameters that are well known to their nominal values in order to prevent them to diverge too far from those values.
- However, most parameters have to be allowed to vary (except for the specified “error” constrains) for the best geometry refinement. So “Reset” for various parameters in “Calibrate” tab is typically not recommended.
- If you do want to reset some parameters to their nominal values, make sure to type the actual reset values. Defaults are all 0!!!
- Start with the same "errors" for parameters used for refinement and adjust if needed.
- Resolution, wavelength range and spots parameters can be the same as for the refinement but can be adjusted if deemed necessary.
- "Rejection" criteria mark which images will not be included in integration due to the fact they have not refined sufficiently, are mis-indexed or contain multiple diffraction patterns (which may affect accurate integration).
- Hoovering the cursor over the fields for “Rejection” provides explanation of various criteria. Good starting numbers for the example data set:
    - 100 - reject images with <100 observed/predicted spots matched
    - 0.8 - reject images with RMSD >0.8
    - 300 - reject images with < 300 total observed spots
    - 20 - reject images with <20% of observed spots matched with predicted spots
    - 10000 - reject images with >10000 observed spots (indicates multiple diffraction patterns on the image)
    - 40 - reject images with > 40 spots in a particular area of the detector; this area is set up when you “View images”, in the “View images” window that opens. Once you set it up, it is shows as a red rectangle when you view an image. You typically want to set this area close to the beamstop where few spots are typically observe. If a large number of spots is present, this indicates multiple diffraction. So this criterion is also used to reject images with multiple diffraction patterns (density of spots is high in the region where density is typically relatively low if a single diffraction pattern is present)
    - Last two criteria help to reject images with 2 or many (multiple) diffraction patterns. I adjusted parameters for the red detector area for the example data set. But this area needs to be adjusted/re-defined for other data sets. The area can be adjusted in the “Calibrate”/”View Images” window as mentioned above.

- Repeat calibrate step 4-5 times for best geometry refinement results. This assures best overlap between observed and predicted spots that is necessary for more accurate integration of spots.
- After repeated calibration steps, click on "Apply Rejections" and check which images got rejected. Play with rejection criteria to adjust what gets rejected. If you change rejection criteria, click on "Apply rejection" to update. Rejections are marked as N (no) for integration in the updated database.txt file.
- Log file calibrate_stats_*.txt, saved after each calibration step, lists which images are "Good" (accepted) and why some are rejected (based on which rejection criteria).
- One can also view and change the status of the images: to be integrated or not. While in the Calibration tab, check Integrate check box. Unclick this check box and click on "Save" for each image that you wants to reject additionally. Or click on this box and save if you do want to include in integration (but was automatically rejected). One can also mark some images as "multiple" if not already automatically marked (which also toggles "integrate" from Y to N).

Files used for and created after refinement:

- For each image, a number of files are used and updated/created:

    - *.mccd.inp - parameters from previous refinement or calibrate step are used as input; this file is also updated/overwritten after each calibrate step
    - *_1_calibrate.inp - input script (commands) for calibrate step created based on the GUI inputs
    - *_1_calibrate.inp.log - log file from calibrate step
    - *.mccd.re.spt - list of found/recognized spots (coordinates)

- *.mccd.pre.spt - list of predicted spots (hkls and coordinates0
- *.mccd.inp - parameters resulting from a calibrate step; used for input for the calibrate step; updated after each calibrate step
- calibrate_stats_*.txt - log file saved after each calibrate step.

Calibration steps for the example data set:

- 436 "Good" images of 486 indexed images based on the above rejection criteria.
- Repeat calibrate step: still 436; checked some images and changed integration status so final number of “Good” images to be integrated, scaled and merged: 437
- Tried changing % of matched spots from 20% to 30% which resulted in 381 "Good" images. But changed back to 20% to include some images with multiple diffraction (so low matched number of spots since only one diffraction pattern was indexed) but not too crowded to affect accurate integration significantly
- Repeat calibrate one more time: 436 "Good" images

==========

Integration

==========

- Use database.txt file to extract images for integration:

Column 1: directory path for images
Column 2: image file name
Column 12: Y if image is marked to be integrated
Column 16: Y if image was marked as a multiple (will not be integrated).

Several steps:

- more database.txt | grep calibrate | awk '{print $2,$12}' > tmp
- grep "mccd Y" tmp > tmp2
- more tmp2 | awk '{print $1}' > images-to-integrate.txt (to strip Y)
- Modify images-to-integrate-newPrecog.txt to add full path to directory with images to file names. Example:
  /mnt/beegfs/data/ihee_2202/slaue/lyso/run5/run5_00175.mccd

Or consolidate all above into one step:
more database.txt | grep calibrate | awk '{print $1,$2,$12}' | grep "mccd Y" | awk '{print $1,$2}' | sed 's/ /\//g' > images-to-integrate.txt

~~~~~~~~~~~~

Example data set:

- images-to-integrate.txt: 438 images to integrate (the rest of indexed images misindexed, rejected based on some criteria like bad geometry refinement or contain many diffraction patterns).

- Used a separate directory for integration, scaling and merging (not necessary, you could use index/refinement directory)

- Copy from the index/refine directory:
  - pyPrecog_settings.py
  - images-to-integrate.txt
  - *mccd.inp - all files

- Need X-ray spectrum - use a measured spectrum file (ask beamline scientists); format is wavelength, relative intensity (see Xray-spectrum_N.lam used for the example data set)
~~~~~~~~~~~~

- Use pyPrecognition.py for integration:
  - Change work directory in the "Setup" tab to the integrate-scale-merge directory
  - In "Integrate" tab, select "Image list from file"
  - Select images-to-integrate.txt as "Images file".
  - Select your spectrum file as “Spectrum file” in the same tab
  - Integration mode: use linearAnalytical (best Precognition integration mode; check Precognition manual for other options)
  - Resolution: set resolution range for integration (2.3,100A used for example data set)
  - Wavelength range: can use same range as for indexing and refinement (1.02-1.18A typical)
  - Spots: can use same numbers as used for indexing and refinement (6 4 3 typical, 3 is the sigma level used for deriving analytical spot profiles)

- Click on “Integrate”.
- *ii files are created with integrated intensities for each image (check Precognition manual for file format).
- Check how many *ii files created (435 of 438 images were successfully integrated for the example data set).
- Create a list of files to scale and merge:
  ls -l *ii | awk '{print "@"$10}' | sed 's/ii/inp/g' > files-to-scale.txt

==========

Scale data

==========

- This is done outside of pyPrecognition by running Epinorm program with a single input file: scale_1.inp
- Cut/paste list of files to scale from the files-to-scale.txt to scale_1.inp
- See Precognition/Epinorm manual for definitions of various parameters for scaling.

Example of scale_1.inp file for the example data set:

```
diagnostic    off
busy          off
warning       on
prompt        off
@run5_00175.mccd.inp
@run5_00176.mccd.inp
@run5_00242.mccd.inp
@run5_00253.mccd.inp
@run5_00256.mccd.inp
@run5_00326.mccd.inp
... (all files to scale)
@run5_13211.mccd.inp
@run5_13360.mccd.inp
@run5_13364.mccd.inp
@run5_13366.mccd.inp
@run5_13374.mccd.inp
prompt        on
Input
  Image   Xray-spectrum_N.lam
  Resolution 2.3 100
  Wavelength 1.02 1.18
  Quit
Scale
  Sigma      3
```

```
   Mosaicity  0 fix
   Isotropy 0 scale
   Isotropy 0 temperature
   Expansion    fix
   Lambda-shift free
   Chebyshev 64
   Mapping   nonlinear
   Mode      global
   Restore   run5_1.inp
Quit
Lambda    run5_1.lam
Quit
```

Scaling command:

```
epinorm scale_1.inp | tee scale_1.log
```

- Statistics for the example data set from scale_1.log file:

```
200736 integrated intensities representing
5096 unique reflections from
435 frames loaded.

-----Initial stats for the example data set:

R-model          = 0.658009
Weighted R-model = 0.559839
R-models calculated from
200520 accepted integrated intensities.
These R-factors indicate how well the integrated
intensities are modeled by the current parameter set.

        R-merge on F^2 = 0.661109
Weighted R-merge on F^2 = 0.598521
        R-merge on F   = 0.33672
Weighted R-merge on F   = 0.295542
R-merges calculated from
200520 accepted integrated intensities of
5096 unique reflections with redundant measurements.
These R-factors indicate how well the symmetry-related
reflections agree with each other.

Mean F^2 / sigma(F^2)   = 7.23186
Mean F   / sigma(F)     = 16.1084
Signal-to-noise ratio calculated from
4864 unique reflections with highly redundant measurements.

-----Final stats for the example data set:

R-model          = 0.0938987
Weighted R-model = 0.0965646
R-models calculated from
163257 accepted integrated intensities.
These R-factors indicate how well the integrated
intensities are modeled by the current parameter set.
```

R-merge on F^2 = 0.116177
Weighted R-merge on F^2 = 0.109147
R-merge on F = 0.0636706
Weighted R-merge on F = 0.059594
R-merges calculated from
163257 accepted integrated intensities of
5027 unique reflections with redundant measurements.
These R-factors indicate how well the symmetry-related
reflections agree with each other.

Mean F^2 / sigma(F^2) = 36.2375
Mean F / sigma(F) = 72.4631
Signal-to-noise ratio calculated from
4750 unique reflections with highly redundant measurements.

X-ray spectrum derived from the example data set (run5_1.lam) seems somewhat shifted compared to the input spectrum (Xray-spectrum_N.lam). This is most likely due to somewhat off cell parameters for a/b that had to be fixed and could not be refined with Precognition. For Laue processing, length scales – wavelength, unit cell and distance are interrelated so it is important to use values for cell parameters and distance that are as accurate as possible.

==========

Merge data

==========

Use apply.inp script. Example below is for the example data set. See Precognition manual for definitions of various merge parameters.

Example of apply.inp file for the example data set:

```
diagnostic    off
busy          off
warning       on
prompt        off
@run5_00175.mccd.inp
@run5_00176.mccd.inp
@run5_00242.mccd.inp
@run5_00253.mccd.inp
@run5_00256.mccd.inp
@run5_00326.mccd.inp
...
@run5_13360.mccd.inp
@run5_13364.mccd.inp
@run5_13366.mccd.inp
@run5_13374.mccd.inp
@run5_1.inp
prompt        on
result        on
Input
  Resolution 2.3 100
  Wavelength 1.02 1.18
  Spot       6 4
  Quit
Scale 3.0 local
  Quit
```

Apply 3 2 run5_3sig_2.3A.hkl
Quit

Command for merging:

epinorm apply.inp | tee apply_3sig.log

- Statistics for the example data set from apply_3sig.log file:

-----Final stats for the example data set:
201809 integrated intensities representing
5274 unique reflections from
435 frames loaded.

R-model = 0.089642
Weighted R-model = 0.093021
R-models calculated from
163219 accepted integrated intensities.
These R-factors indicate how well the integrated
intensities are modeled by the current parameter set.

R-merge on F^2 = 0.106308
Weighted R-merge on F^2 = 0.101594
R-merge on F = 0.0592867
Weighted R-merge on F = 0.0563826
R-merges calculated from
163160 accepted integrated intensities of
5023 unique reflections with redundant measurements.
These R-factors indicate how well the symmetry-related
reflections agree with each other.

Mean F^2 / sigma(F^2) = 38.8749
Mean F / sigma(F) = 77.7236
Signal-to-noise ratio calculated from
4738 unique reflections with highly redundant measurements.

Resolution range (A)	Unique refl.	Mean F^2/sigma(F^2)	Mean F/sigma(F)
1000.0000 - 5.8042	373	35.47	70.74
5.8042 - 4.6068	348	40.68	81.00
4.6068 - 4.0244	358	53.04	105.88
4.0244 - 3.6564	332	54.14	108.21
3.6564 - 3.3943	331	51.56	102.86
3.3943 - 3.1942	332	47.50	94.87
3.1942 - 3.0342	333	43.72	87.56
3.0342 - 2.9021	308	41.28	82.87
2.9021 - 2.7904	320	38.89	78.12
2.7904 - 2.6941	306	36.79	73.76
2.6941 - 2.6098	299	33.74	67.65
2.6098 - 2.5352	298	30.52	61.11
2.5352 - 2.4685	300	25.75	51.46
2.4685 - 2.4082	259	20.95	41.80
2.4082 - 2.3535	187	15.31	30.45
2.3535 - 2.3034	54	15.48	30.83

Single reflections:

Resolution (A)		Unique	Observed	Completeness (%)	
100.00 -	4.60	762	9864	99.16	
4.60 -	3.65	710	9988	99.44	
3.65 -	3.19	688	9936	98.73	
3.19 -	2.90	666	9682	98.41	
2.90 -	2.69	670	9852	97.51	
2.69 -	2.53	658	9748	96.94	
2.53 -	2.40	635	9432	94.13	
2.40 -	2.30	456	6928	69.60	
100.00 -	2.30	5245	75430	94.25	

Singles+multiples (after harmonic deconvolution – see Precognition manual):

Resolution (A)		Unique	Observed	Completeness (%)	
100.00 -	4.60	764	9876	99.28	
4.60 -	3.65	712	9996	99.52	
3.65 -	3.19	690	9944	98.81	
3.19 -	2.90	669	9694	98.54	
2.90 -	2.69	674	9888	97.86	
2.69 -	2.53	659	9752	96.98	
2.53 -	2.40	637	9444	94.25	
2.40 -	2.30	458	6952	69.84	
100.00 -	2.30	5263	75546	94.40	

Output file: *.hkl
Format: h,k,l, F, sigF (Fs, not intensities!!!), can strip and ignore last two columns)