

XChemExplorer

Manual

Current version: v1.3

Date

11/01/2019

Author

Tobias Krojer

tobias.krojer@sgc.ox.ac.uk

Table of Contents

Introduction	4
Reference	4
Getting started	5
Prerequisites	5
Installation	5
Usage	6
Compatibility	6
Notes	6
Acknowledgements	6
Settings	7
Preferences	9
Analysing DLS auto-processing results	10
Beamline directory	10
Getting data into XCE	10
Manually selecting auto-processing results	10
Changing selection criteria	10
Rescoring	11
Known issues	11
Database	12
Database update	12
Dataset reprocessing	17
Running xia2 or dials	17
Getting the results into the database	19
Initial Map Calculation	21
Twining	25
Ligand Restraints	26
PanDDA	27
Suggested Workflow	27
Ground-state model building	27
Initial reference model selection and quick preparation	28
Calculating dimple maps and models in XCE	28
PanDDA pre-run	28
Ground state model building	30
Map re-calculation	32
pandda.analyse	33
pandda.inspect	33
PanDDA model export to the project directory	33
Convert Event Maps to MTZ files	33
Refinement	35
Refinement stages	35

Overview	36
Deposition.....	40
Adding compulsory information	40
Preparation of files for PDB upload.....	42
Tips & Tricks	44
DLS remote connection.....	44
Using XChemExplorer outside the labxchem environment.....	46
Project Directory structure	46
Updating the database	48
Running PanDDA	48
Refinement.....	48
Deposition	48
Frequently asked questions.....	49
What happens when you step through the models?	49
What happens when I make changes to the model?	49
What happens when I refine the model?	49
I have wiped the contents of the project (initial_model) directory, but '<i>Get new results from autoprocessing</i>' does nothing when I run it again	50
Known Problems	51

Introduction

XChemExplorer (XCE) is a data management and workflow tool, which supports large-scale analysis of protein-ligand structures by X-ray crystallography. It is not an actual algorithm, but serves as a launch pad for batch submission and analysis of the essential steps in the structure determination of protein-ligand structures.

Reference

Krojer, T., Talon, R., Pearce, N., Collins, P., Douangamath, A., Brandao-Neto, J., Dias, A., Marsden, B., and von Delft, F. (2017). The XChemExplorer graphical workflow tool for routine or large-scale protein–ligand structure determination. *Acta Cryst D* 73, 267–278.

XChemExplorer makes extensive use of other people's software, therefore please cite their work accordingly:

- XIA2
- DIMPLE
- ACEDRG
- PanDDA
- COOT
- REFMAC
- PHENIX
- MolProbity
- GRADE

Getting started

Prerequisites

XCE works on any Mac OSX or Linux system, but it is essential that CCP4 (www ccp4.ac.uk) version 7.0 or higher is installed and correctly configured. XCE uses the python version that comes with it and will therefore not work if it does not exist. Additionally, it may be useful to install PHENIX, since XCE uses several of its tools for validation purposes.

Installation

Download XChemExplorer from <http://tkrojer.github.io/XChemExplorer>

Put the gzipped tar archive to wherever you want XCE to be installed. In case you have no root privileges, put it somewhere into your home directory, e.g.:

/home/tkrojer/software

Then change to the respective directory and unpack the archive, e.g.:

```
cd /home/tkrojer/software  
gunzip XChemExplorer-1.2.tar.gz  
tar -xvf XChemExplorer-1.2.tar
```

This will create a new directory, i.e. the XChemExplorer directory. Change into this directory, e.g.:

```
cd XChemExplorer-1.2
```

The contents of the directory should look something like this when you type '**ls -l**':

```
-rwxr-xr-x 1 tkrojer users 238K Jun 18 13:58 XChemExplorer.py  
-rwxr-xr-x 1 tkrojer users 269 Jun 18 13:58 XChemExplorer_local.sh  
-rwxr-xr-x 1 tkrojer users 316 Jun 18 13:58 XChemExplorer_dmd.sh  
drwxr-xr-x 3 tkrojer users 4.0K Jun 18 13:58 web  
-rwxr-xr-x 1 tkrojer users 553 Jun 18 13:58 setupssh.sh  
drwxr-xr-x 2 tkrojer users 4.0K Jun 18 13:58 setup-scripts  
-rwxr-xr-x 1 tkrojer users 2.8K Jun 18 13:58 README.md  
drwxr-xr-x 2 tkrojer users 4.0K Jun 18 13:58 lib  
drwxr-xr-x 2 tkrojer users 4.0K Jun 18 13:58 image  
drwxr-xr-x 2 tkrojer users 4.0K Jun 18 13:58 icons  
drwxr-xr-x 2 tkrojer users 4.0K Jun 18 13:58 helpers  
drwxr-xr-x 2 tkrojer users 4.0K Jun 18 13:58 gui_scripts  
-rwxr-xr-x 1 tkrojer users 182 Jun 18 13:58 Dockerfile  
-rwxr-xr-x 1 tkrojer users 465 Jun 18 13:58 compile_test.py  
lrwxrwxrwx 1 tkrojer users 20 Jun 18 14:33 XChemExplorer -> XChemExplorer_dmd.sh
```

The only thing left to do is to edit the XChemExplorer_dmd.sh file with a text editor. Change the line

```
export XChemExplorer_DIR='/usr/local/scripts/tobias/XChemExplorer'
```

to where you XChemExplorer is installed. In our example this would be

```
export XChemExplorer_DIR='/home/tkrojer/software/XChemExplorer-1.2'
```

That's it!

Limitations: XCE does currently not have a dedicated update mechanism. It is possible to clone the current master branch of the XCE github repository (<https://github.com/xchem/XChemExplorer>) and then frequently pull from it!

Usage

You can now run XCE by typing

```
/home/tkrojer/software/XChemExplorer-1.2/XChemExplorer_dmd.sh
```

It may however be easier if you insert an alias into your .bashrc or .cshrc file:

```
alias XChemExplorer='/home/tkrojer/software/XChemExplorer-1.2/XChemExplorer.sh'
```

Compatibility

XCE does not have a problem with compatibility between different versions. This is because the program is essentially file system centred. Please see the original publications for more information.

Notes

The terms *datasource* and *database* are used somewhat interchangeably in the XCE GUI and in this manual.

Acknowledgements

XCE was originally written by me, but the program is currently being co-developed and maintained together with Rachael Skyner. Elliot Nelson contributed significantly to the design and development of the PanDDA refinement module. Frequent feedback from our amazing colleagues at the SGC and Diamond Light Source and input from the XChem user community was of utmost importance for software development. Special thanks goes to Romain Talon, Patrick Collins, Alice Douangamath, Jose Brandao-Neto and Alexandre Dias. The Research Informatics team at the SGC was instrumental for implementing the generation of HTML summary pages. The team at RCSB Rutgers have massively helped to support the group deposition mechanism. Finally, this work would not have been possible without the generous support from various SGC funders: the SGC is a registered charity (No. 1097737) that receives funds from AbbVie, Bayer, Boehringer Ingelheim, the Canada Foundation for Innovation, the Canadian Institutes for Health Research, Genome Canada, GlaxoSmithKline, Janssen, Lilly Canada, the Novartis Research Foundation, the Ontario Ministry of Economic Development and Innovation, Pfizer, Takeda and the Wellcome Trust (092809/Z/10/Z).

Settings

The settings tab (Figure 1) contains information where XChemExplorer (XCE) will write files to and where it can find certain files. It also tells XCE the name and location of the SQLite database file.

If you use XCE at DLS as part of an XChem project and open it somewhere in /dls/labxchem/... then you will usually not have to change anything. XCE will populate the directories and information about the SQLite database file with defaults.

The situation is different when you leave the labxchem environment; now you need to specify the information. However, there are only 3 pieces of information required to get started:

Project directory

This is where all files that XCE creates will end up. The project directory contains a sub-directory for every crystal and the name of every sub-directory corresponds to the *CrystalName* field in the database. Please note that the names can be completely arbitrary, although it is highly recommended to choose meaningful names. Please, check the XCE publication for more information about the structure and filename conventions of the project directory.

Reference Structure Directory

You can provide reference PDB, MTZ and CIF files in this directory. These files will be used for map calculation with DIMPLE or for selection of the best auto-processing result. The filenames can be arbitrary, however, if the files belong together they need to have the same root.

Data Source

Here you need to specify the database file that you want to use. If you are starting from scratch, please check the next section, which explains how to create a new database file

CCP4_SCR director

This is the same as your CCP4_SCR directory. It is essentially a scratch directory, which XCE uses to save input scripts for processing, restraints generation and refinement to. Usually one can ignore this directory, however, it is a good place to start trouble-shooting in case a job did not behave as expected.

Group Deposition Directory

This is the directory where XCE will write the gzipped tar archive file for upload into the Protein Data Bank.

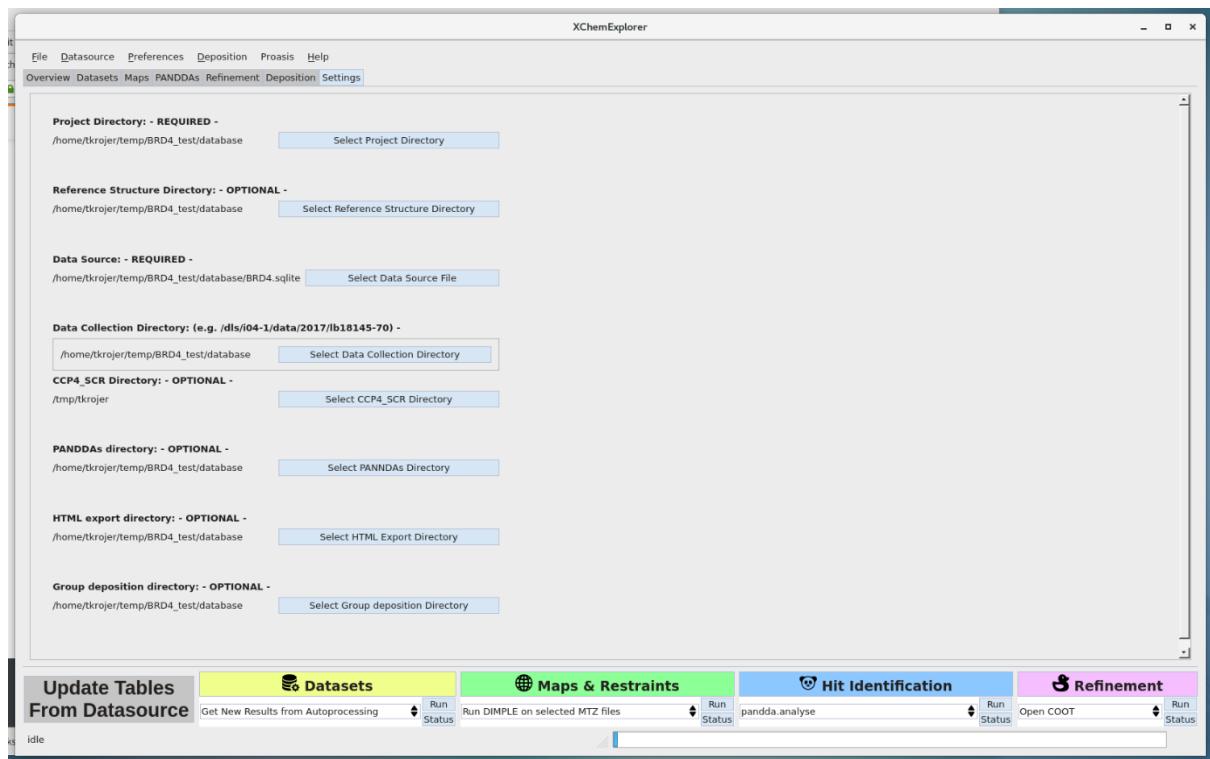


Figure 1. The settings tab in XChemExplorer.

Preferences

--- coming soon ---

Analysing DLS auto-processing results

XCE will read, copy and analyse the auto-processing results from data collected at the Diamond Light Source. Initially, it will check if the respective files exist, then it will copy the final MTZ and LOG files to the autoprocessing folder in the respective sample directory and finally it will try to select the ‘best’ result for further analysis, i.e. initial refinement. Please check out the XCE paper in case you want to know what ‘best’ means. It does currently read the results from xia2, dials, autoProc and autoProc-staraniso. Please note that the results from the fast_dp pipeline are ignored from v1.2 onwards! Additionally, It is important that the data are collected like this:

<protein_name>/<sample_name>

e.g.

lysozyme/lysozyme-x0001

Beamline directory

First, you need to select the *actual* data collection visit directory in the settings tab. Please ignore the beamline directory in your labxchem folder! In the past one could link multiple visits into the beamline directory, but this feature is obsolete!

XXX picture missing XXX

Getting data into XCE

Then, switch to the Datasets tab and select your target from the dropdown menu.

XXX picture missing XXX

Select ‘Get New Results from Autoprocessing’ from the yellow action box and press ‘Run’.

XXX picture missing XXX

Manually selecting auto-processing results

--- missing ---

Changing selection criteria

--- missing ---

Rescoring

--- missing ---

Known issues

There are sometimes cases where you may have made changes to the project directory, but when you want to parse the data collection visit directory again, nothing happens. See I have wiped the contents of the project (initial_model) directory, but '*Get new results from autoprocessing*' does nothing when I run it again for details.

Database

XCE uses a simple SQLite database to capture information, results and outcomes that are generated during the project. If your samples were collected at the XChem facility at the Diamond Light Source, then you will usually just take the soakDB file that was created during crystal preparation. Otherwise, select '*Create New Datasource (SQLite)*' from the *Datasource* menu and create a new database (Figure 2).

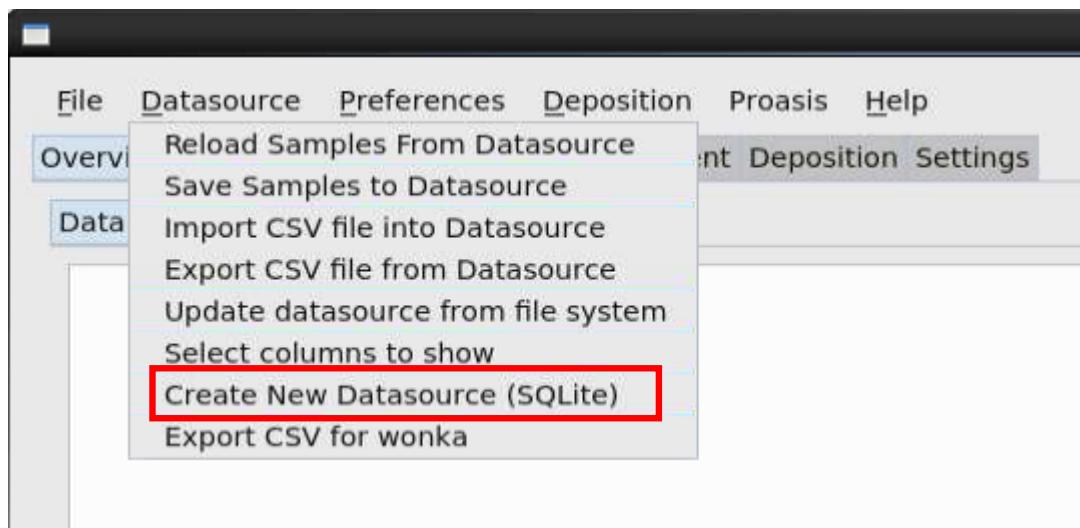


Figure 2. Datasource menu item '*Create New Datasource (SQLite)*'.

Database update

There should be no need to make changes to the database when working on an XChem project. One notable exception is the entry of information about soaked/ co-crystallised compounds in case these compound are not part of the XChem fragment libraries. The initial ambition was to have a database update functionality available as part of XCE, but this approach will not be pursued for the time being. The *Datasource* menu contains some rudimentary functionalities for database changes, but it is recommended to follow the instructions outlined here in case the need for database manipulation arises. Also, keep in mind that the tables in XCE only display the contents of the database. Changing the value of the fields, although it is possible, has not effect on the database! A single click on '*Update Table From Datasource*' will bring back the actual content of the database.

Software

There are several programs for manipulation of SQLite files, the one we typically use is SQLite Browser (<https://sqlitebrowser.org>) (Figure 3). It is available for Windows, Mac and Linux.

Instructions

It is easily possible to make all the required changes directly within SQLite Browser, but this really only make sense if you need to change a few fields. For large-scale data entry, you should export the

respective table (usually only the *mainTable*) into CSV format and then make the required changes in Microsoft Excel. Other programs like OpenOffice should work as well, however, there have been anecdotal reports of import problems.

Before you start, make a copy of the current database in case something goes wrong!!!

ID	CrystalName	ProteinName	aCollectionVis	aCollectionRun	aCollectionBeam	aCollectionOutcc	aCollectionDat	Colle	Collection Name
1	NULL	NUDT7A-x1711	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
2	NULL	NUDT7A-x1711	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
3	NULL	NUDT7A-x1711	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
4	NULL	NUDT7A-x1711	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
5	NULL	NUDT7A-x1711	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
6	NULL	NUDT7A-x1711	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
7	NULL	NUDT7A-x1711	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
8	NULL	NUDT7A-x1711	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
9	NULL	NUDT7A-x1712	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
10	NULL	NUDT7A-x1712	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
11	NULL	NUDT7A-x1712	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
12	NULL	NUDT7A-x1712	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
13	NULL	NUDT7A-x1713	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
14	NULL	NUDT7A-x1713	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
15	NULL	NUDT7A-x1713	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
16	NULL	NUDT7A-x1713	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
17	NULL	NUDT7A-x1714	NULL	mx15433-43	NUDT7A-x171...	i24	NULL	NULL	NULL
18	NULL	NUDT7A-x1714	NULL	mx15433-43	NUDT7A-x171...	i24	NULL	NULL	NULL
19	NULL	NUDT7A-x1714	NULL	mx15433-43	NUDT7A-x171...	i24	NULL	NULL	NULL
20	NULL	NUDT7A-x1714	NULL	mx15433-43	NUDT7A-x171...	i24	NULL	NULL	NULL
21	NULL	NUDT7A-x1715	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
22	NULL	NUDT7A-x1715	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
23	NULL	NUDT7A-x1715	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96
24	NULL	NUDT7A-x1715	NUDT7A	mx15433-43	NUDT7A-x171...	i24	success	2017-12-19 0...	0.96

Figure 3. SQLite browser main window.

Next, open the sqlite file and export the export *mainTable* to a CSV file:

File -> Export -> Table(s) as CSV file...

Save as *mainTable.csv* if asked. Make sure that the box ‘*Column names in first line*’ is checked (Figure 4).

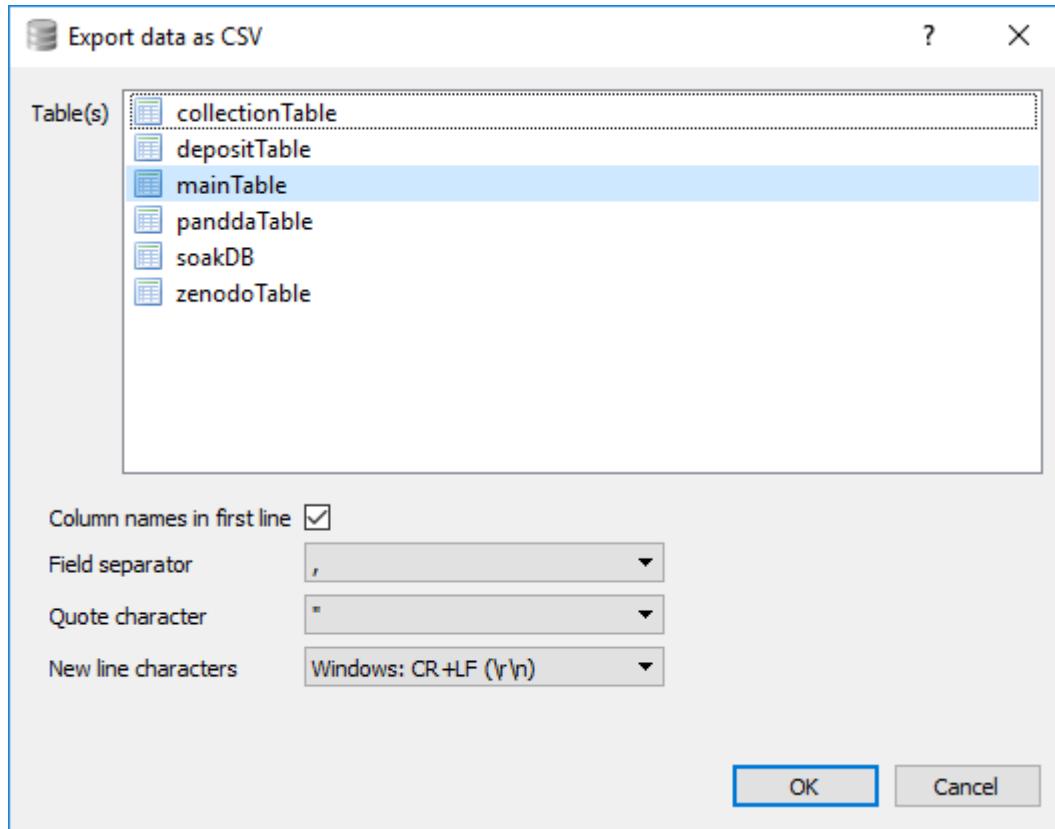


Figure 4. SQLite browser - export table to CSV file.

Next, delete the *mainTable* from database. In the ‘Database Structure’ tab, highlight the table you want to delete, then, go to Edit -> Delete Table (Figure 5). Press ‘Write Changes’ and close the file.

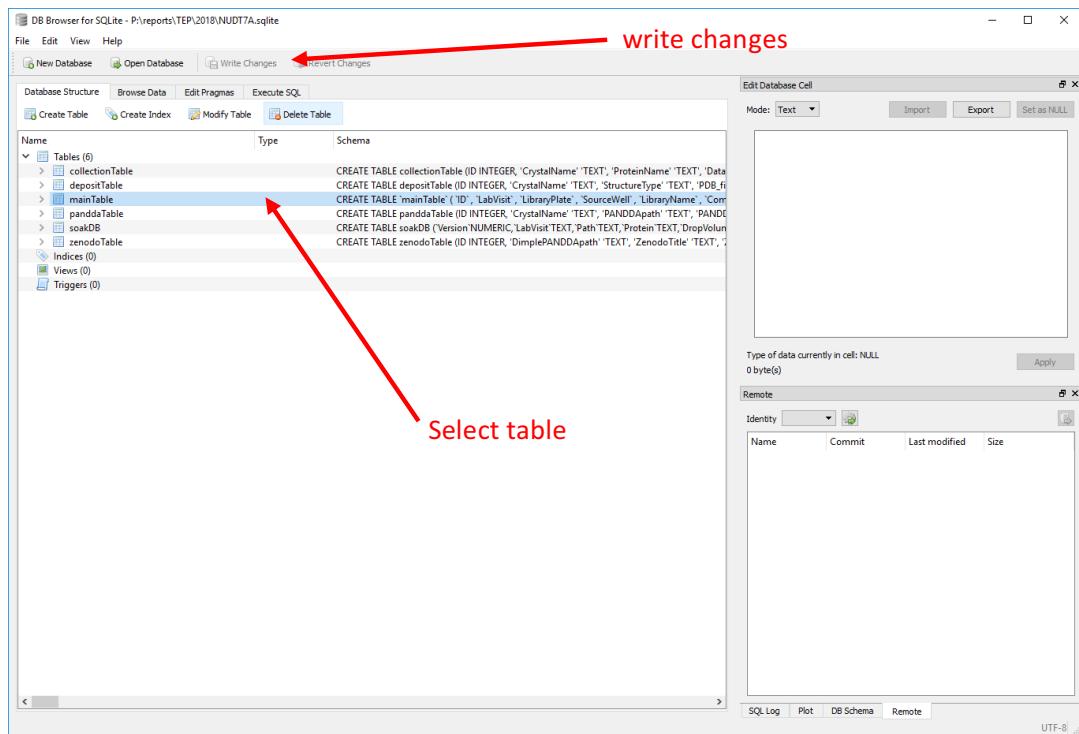


Figure 5. SQLite browser - select table for deletion.

Now open the CSV file in Microsoft EXCEL and make changes as necessary. The fields that most often will need changing are *CompoundSMILES* and *CompoundCODE*.

Open the SQLite file again and select File -> Import -> Table from CSV file...

Make sure the table name is *mainTable* and that '*Column names in first line*' is checked (Figure 6).

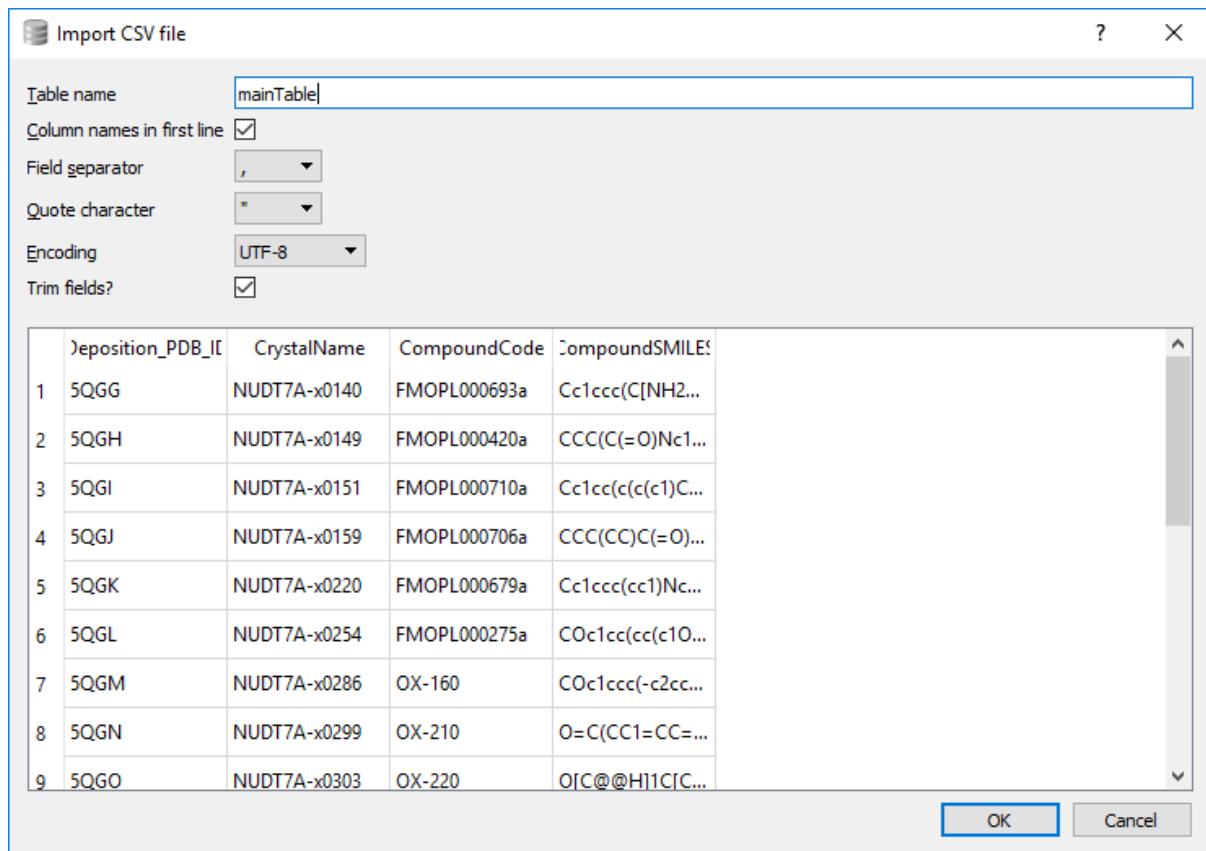


Figure 6. SQLite browser - import CSV file as table into SQLite database.

Dataset reprocessing

Running xia2 or dials

XCE can be used to reprocess datasets with either XIA2 or DIALS. This option is available in the Datasets/ Reprocess tab (Figure 7).

First, select the datasets directory, then, press ‘*Search Datasets*’. Note that if you want to process multiple-datasets then select the respective top-level directory under which the individual datasets are stored.

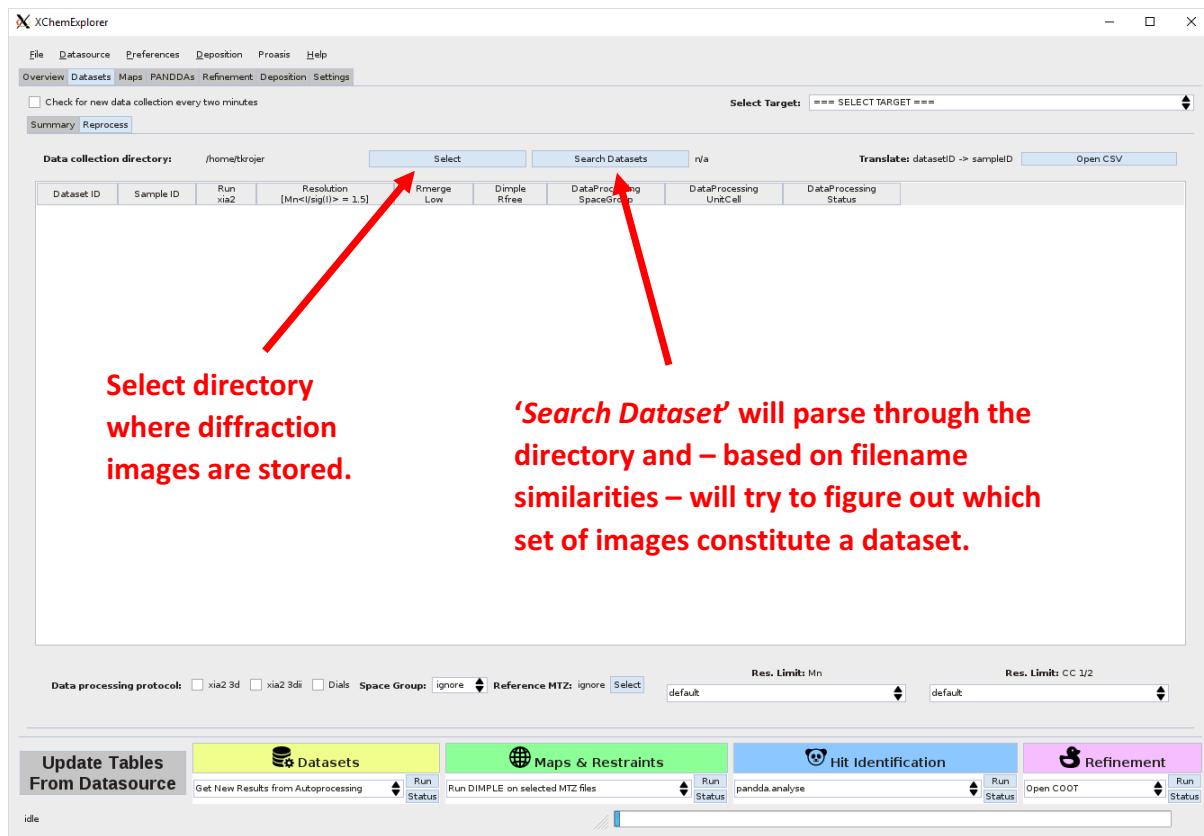


Figure 7. (Re-) processing of diffraction data in XCE.

Depending on the number of datasets, parsing of the respective directory may take a while. However, the progress bar will indicate how far along the program is and finally the table should get populated as in the exemplary screenshot below. There are a few things one should keep in mind:

- the *datasetID* is the name of each folder in the ‘*Data collection directory*’; e.g. all subfolders in /xdata/BRD4A
- only folders with more than 20 diffraction images in them will be listed

- XCE assumes that *datasetID=sampleID*; and if there is already an entry in the datasource for a particular *sampleID* then it will display things like Rmerge etc.
- you can change the sampleID simply by entering another value in the cell (Figure 8)
- if you want to use different sample IDs for many datasets, then you can use the '*Translate dataset ID -> sample ID*' option.
- you can either select all samples you want to run by clicking on the checkbox or you select a range of rows with the mouse, right-click, click on 'mark selected for reprocessing'

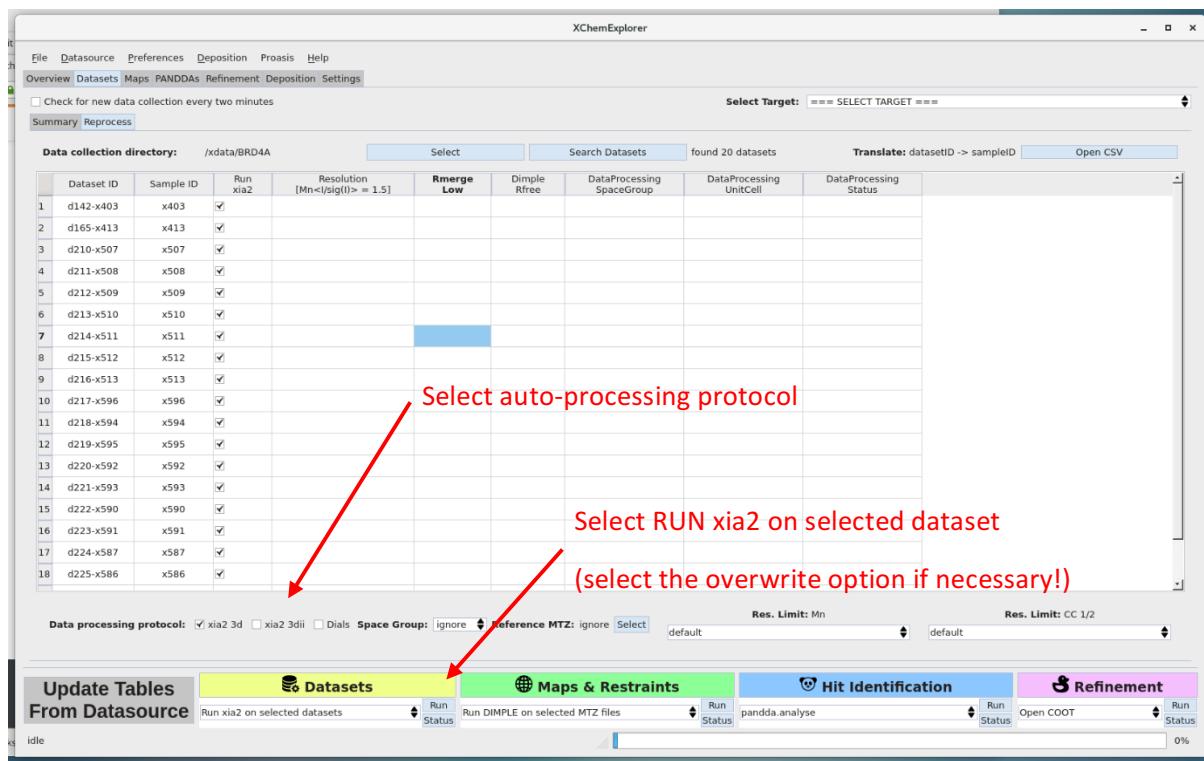


Figure 8. Dataset reprocessing tab after XCE found and assigned the diffraction images. Note that the values in the Sample ID column were manually entered. It would have been the same as in the Dataset ID column when XCE populated the table the first time.

Next, select the data processing protocol(s) you want to use (red arrow above), then select '*Run xia2 on selected datasets*' from the yellow *Datasets* actionbox and press '*RUN*'. Note that all data processing results will end up in the project directory.

If you want to provided reference files then put the respective files into the reference directory.

Please note that if your local machine is not automatically able to submit jobs to a computer cluster via qsub, it will process the datasets sequentially on the local machine. Needless to say that this may keep the machine busy for a while in case you want to process tens or even hundreds of datasets. In this case, it is advisable to use only one data processing protocol.

Please also note that XCE does currently not indicate when the jobs are finished (v1.2). You need to check the workload on your machine to find out.

Getting the results into the database

Now you need to get the data processing results/ outcomes back into the database. Stay in the main *Datasets* tab, but switch to the *Summary* sub-tab (Figure 9). Go to the select targets dropdown and select ‘*==== project directory ===*’. Select ‘*Get New results from Autoprocessing*’ from the yellow action box and press *RUN*.

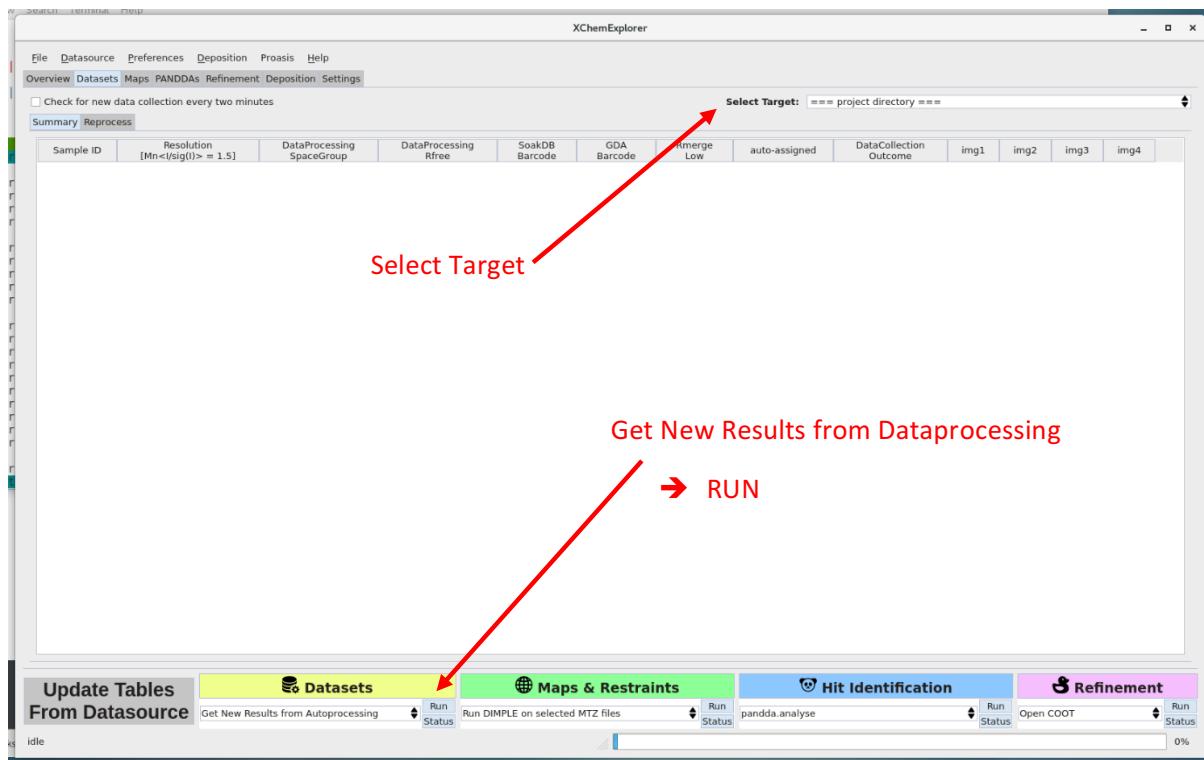


Figure 9. (Re-) processing of diffraction data in XCE. Reading the results.

If everything worked well, each sample directory with a successful data processing outcome should look something like this:

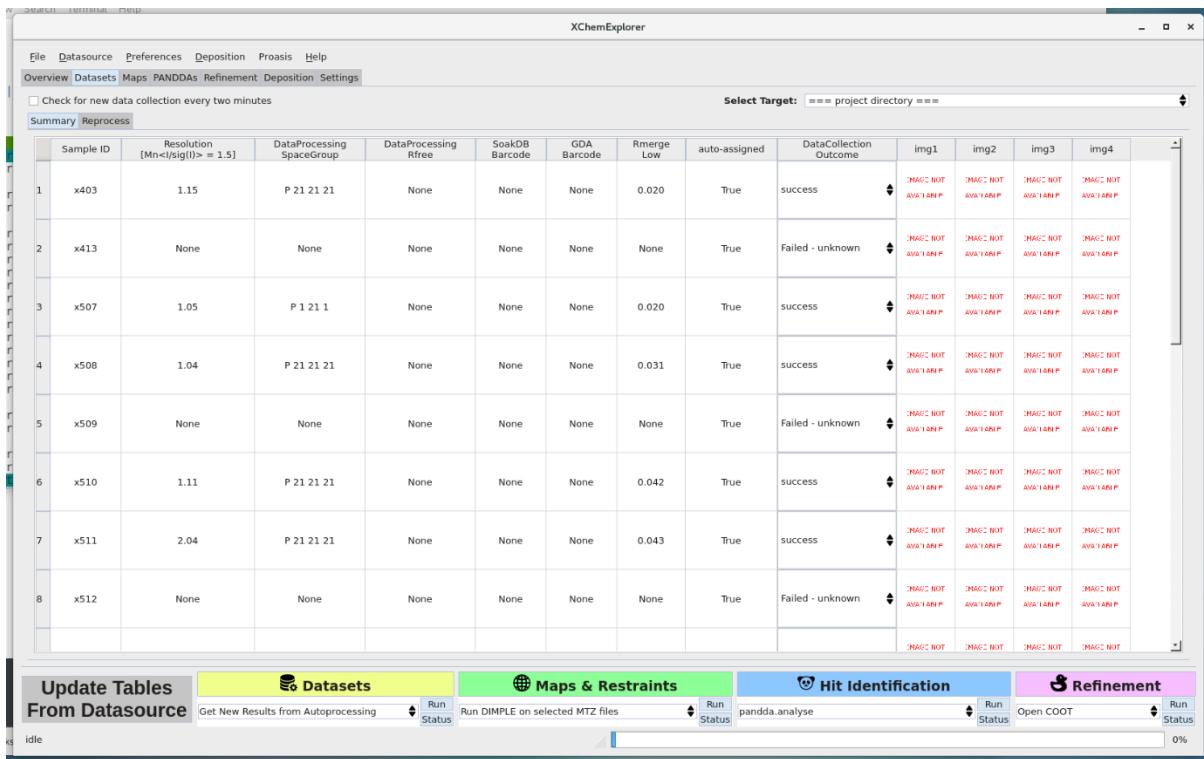


Figure 10. Results from reproccsing with xia2.

You can check the respective sample directories; as you can see below, there should be symbolic links to the MTZ and AIMLESS logfile from xia2; and the files have the same root as the respective sample directory.

```
drwxr-xr-x 3 tkrojer users 4.0K Jun 18 13:37 diffraction_images
drwxr-xr-x 3 tkrojer users 4.0K Jun 18 14:42 processed
drwxr-xr-x 3 tkrojer users 4.0K Jun 18 15:50 jpg
drwxr-xr-x 3 tkrojer users 4.0K Jun 18 15:50 autoprocessing
lrwxrwxrwx 1 tkrojer users   58 Jun 18 15:50 x596.mtz -> autoprocessing/database-run_13d/AUTOMATIC_DEFAULT_free.mtz
lrwxrwxrwx 1 tkrojer users   61 Jun 18 15:50 x596.log -> autoprocessing/database-run_13d/AUTOMATIC_DEFAULT_aimless.log
```

Initial Map Calculation

XCE uses DIMPLE to perform an initial round of refinement and to calculate the resulting 2fofc and fofc maps.

Please note that the tables in XCE do no update automatically! It does so during startup, but they will not refresh automatically. Also, there is currently a bug which means that the status indicators in the MAPS table are not always up to date (v1.2).

If you have successfully processed your datasets or read in the results from DLS auto-processing, press on the grey '*update tables from datasource*' button on the lower left hand corner (Figure 11). You can press this button as often as you want and it will update all tables with the exception being the Datasets Summary table with the latest information from the database.

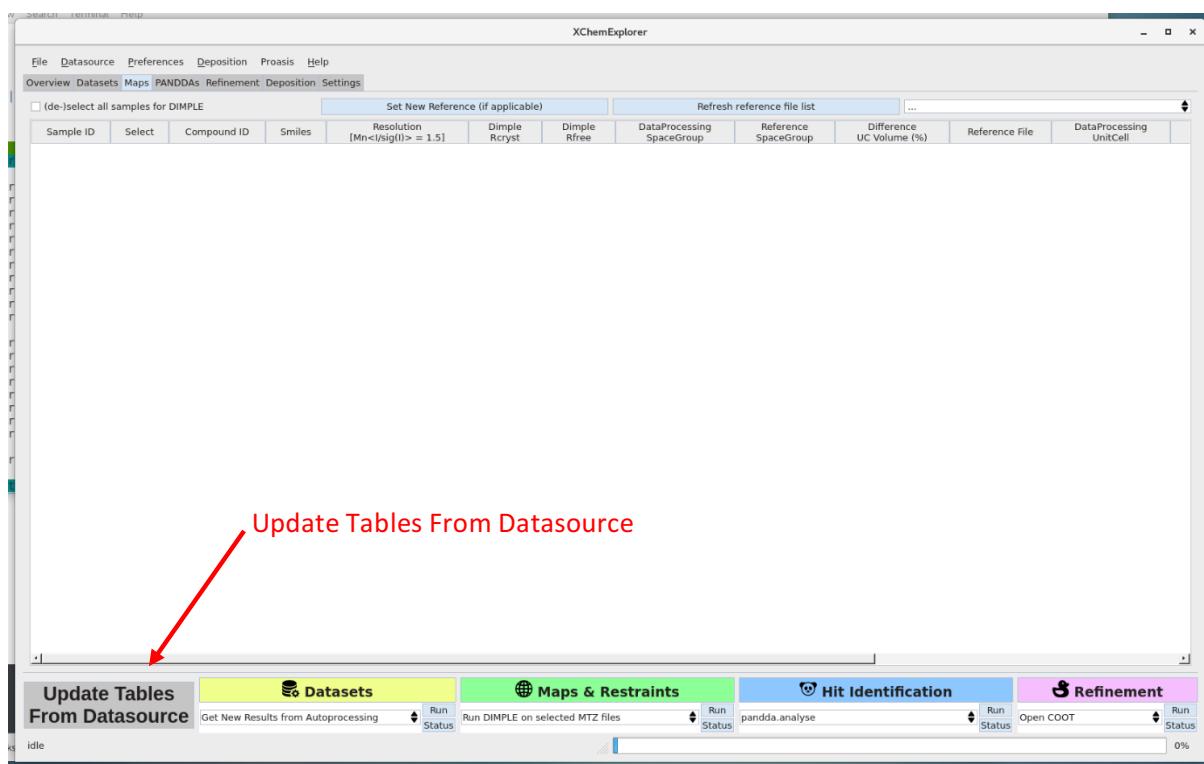


Figure 11. Update tables from datasource. All tables in XCE are currently not updated automatically and need active pulling from the database.

In our current example, the table will look like this:

The screenshot shows the XChemExplorer software interface. At the top, there is a menu bar with File, Datasource, Preferences, Deposition, Proasis, Help, Overview, Datasets, Maps, PANDDAs, Refinement, Deposition, and Settings. Below the menu is a toolbar with buttons for Update Tables From Datasource, Datasets, Maps & Restraints, Hit Identification, and Refinement. The main area contains a table titled 'Set New Reference (if applicable)' with columns: Sample ID, Select, Compound ID, Smiles, Resolution [$Mn<|l/\sigma(l)|> = 1.5$], Dimple Rcryst, Dimple Rfree, DataProcessing SpaceGroup, Reference SpaceGroup, Difference UC Volume (%), Reference File, and DataProcessing UnitCell. The table has 15 rows, each corresponding to a sample ID from x403 to x596. The 'Reference File' column for all samples is currently empty, indicated by three dots (...). The 'DataProcessing UnitCell' column also shows empty entries. The bottom right of the table shows a progress bar at 0%.

Set New Reference (if applicable)											Refresh reference file list	...
	Sample ID	Select	Compound ID	Smiles	Resolution [$Mn< l/\sigma(l) > = 1.5$]	Dimple Rcryst	Dimple Rfree	DataProcessing SpaceGroup	Reference SpaceGroup	Difference UC Volume (%)	Reference File	DataProcessing UnitCell
1	x403	<input type="checkbox"/>	None	None	1.15	None	None	P 21 21 21		999.0	...	37 44 78 90 90 90
2	x507	<input type="checkbox"/>	None	None	1.05	None	None	P 1 21 1		999.0	...	7 32 15 90 96 90
3	x508	<input type="checkbox"/>	None	None	1.04	None	None	P 21 21 21		999.0	...	37 43 78 90 90 90
4	x510	<input type="checkbox"/>	None	None	1.11	None	None	P 21 21 21		999.0	...	37 44 78 90 90 90
5	x511	<input type="checkbox"/>	None	None	2.04	None	None	P 21 21 21		999.0	...	37 44 79 90 90 90
6	x513	<input type="checkbox"/>	None	None	1.29	None	None	P 21 21 21		999.0	...	37 43 77 90 90 90
7	x586	<input type="checkbox"/>	None	None	1.00	None	None	P 21 21 21		999.0	...	37 44 78 90 90 90
8	x587	<input type="checkbox"/>	None	None	1.45	None	None	P 1 2 1		999.0	...	55 41 59 90 103 90
9	x588	<input type="checkbox"/>	None	None	1.82	None	None	P 1		999.0	...	30 39 66 92 98 90
10	x589	<input type="checkbox"/>	None	None	1.00	None	None	P 21 21 21		999.0	...	37 43 78 90 90 90
11	x592	<input type="checkbox"/>	None	None	1.30	None	None	P 21 21 21		999.0	...	37 44 78 90 90 90
12	x593	<input type="checkbox"/>	None	None	1.13	None	None	P 21 21 21		999.0	...	37 44 78 90 90 90
13	x594	<input type="checkbox"/>	None	None	1.00	None	None	P 21 21 21		999.0	...	37 44 78 90 90 90
14	x595	<input type="checkbox"/>	None	None	1.00	None	None	P 21 21 21		999.0	...	37 44 78 90 90 90
15	x596	<input type="checkbox"/>	None	None	1.02	None	None	P 21 21 21		999.0	...	37 44 78 90 90 90

Figure 12. Populated Maps table.

We can now see information about the high resolution limit and space group for each crystal in the table, but the information about which reference PDB file to use is still empty. This is because in the example I have not yet provided a suitable reference PDB file in the reference directory. After this is done, one needs to press '*Set New Reference (if applicable)*', then select the reference file that you want to use, and press '*Run*'.

The screenshot shows the XChemExplorer interface with the 'Maps & Restraints' tab active. The main area is a table of 15 samples. The columns are: Sample ID, Select, Compound ID, Smiles, Resolution [$Mn < l/sig(l) > = 1.5$], Dimple Rcryst, Dimple Rfree, DataProcessing SpaceGroup, Reference SpaceGroup, Difference UC Volume (%), Reference File, and DataProcessing UnitCell. The 'Select' column contains checkboxes. The 'Reference File' column shows file paths like '37 44 78 90 90 90'. The 'DataProcessing UnitCell' column shows values like '37 43 78 90 90 90'. Action buttons at the bottom include 'Update Tables From Datasource', 'Datasets' (with 'Get New Results from Autoprocessing' and 'Run Status'), 'Maps & Restraints' (with 'Run DIMPLE on selected MTZ files' and 'Run Status'), 'Hit Identification' (with 'pandda.analyse' and 'Run Status'), and 'Refinement' (with 'Run COOT' and 'Run Status').

Figure 13. Maps table after reference files for DIMPLE have been assigned.

You can check the XCE publication for more information about how XCE selects which reference file to use. Briefly, XCE will go through all PDB files that you have provided in the reference directory and read the CRYST card in their header. From the CRYST card, it will calculate the unit cell volume and determine the point group. Then it will compare this to the unit cell volume and point group of your crystal. If the unit cell volume differs by less than 12% (this can be adjusted in the *Preferences* menu) and the two have the same point group, then XCE will consider this to be a suitable input file for Dimple. This selection mechanism works well as long as you have either different point groups or the unit cell volume differs significantly between different space groups. It will most likely struggle in cases where you have different crystal forms that are not detectable by the rather coarse selection mechanism! However, I have often found that if XCE does not find a suitable reference file it is recommended to manually check what is going on. It may be that the presence of a certain ligand triggered some change that is best investigated outside the XCE workflow. If it really is a different crystal form, solve the structure as you usually would, then add the resulting structure to your reference file directory.

Now select which crystals you want to process (or simply choose *select all samples for DIMPLE*) and then choose '*Run DIMPLE on selected MTZ files*' from the green action box and press '*RUN*'.

(De-) select ALL or specific samples

Get New Results from Autoprocessing → RUN

If everything worked as expected, then the respective sample directory should now look like this:

```
drwxr-xr-x 3 tkrojer users 4.0K Jun 18 13:37 diffraction_images
drwxr-xr-x 3 tkrojer users 4.0K Jun 18 15:15 processed
drwxr-xr-x 3 tkrojer users 4.0K Jun 18 15:50 jpg
drwxr-xr-x 3 tkrojer users 4.0K Jun 18 15:50 autoprocessing
lrwxrwxrwx 1 tkrojer users 58 Jun 18 15:50 x403.mtz -> autoprocessing/database-run_13d/AUTOMATIC_DEFAULT_free.mtz
lrwxrwxrwx 1 tkrojer users 61 Jun 18 15:50 x403.log -> autoprocessing/database-run_13d/AUTOMATIC_DEFAULT_aimless.log
drwxr-xr-x 3 tkrojer users 4.0K Jun 18 16:18 dimple
lrwxrwxrwx 1 tkrojer users 53 Jun 18 16:21 dimple.pdb -> dimple/dimple_rerun_on_selected_file/dimple/final.pdb
lrwxrwxrwx 1 tkrojer users 53 Jun 18 16:21 dimple.mtz -> dimple/dimple_rerun_on_selected_file/dimple/final.mtz
lrwxrwxrwx 1 tkrojer users 53 Jun 18 16:21 2fofc.map -> dimple/dimple_rerun_on_selected_file/dimple/2fofc.map
lrwxrwxrwx 1 tkrojer users 52 Jun 18 16:21 fofo.map -> dimple/dimple_rerun_on_selected_file/dimple/fofc.map
-rw-r--r-- 1 tkrojer users 1.3M Jun 18 16:21 x403.free.mtz
```

You can now either run PanDDA or move straight to the Refinement tab and look at your electron density maps after initial refinement.

Twinning

If your data are almost perfectly twinned, chances are high that the autoprocessing pipelines will merge the data in the apparent higher symmetry laue group. Once this has happened, you need to reprocess the data in the respective lower symmetry laue group.

Ligand Restraints

--- coming soon ---

PanDDA

XCE can be used to conveniently launch the PanDDA software. However, this is not necessary! One can also run *pandda.analyse* and *pandda.inspect* from the command line and only later, during the *pandda.export* stage use XCE to bring the results back into the XCE folder structure for further refinement. Also, keep in mind that PanDDA does not know about the XCE database! PanDDA only looks at the files in the specified data directory, which is the XCE project directory. The actual PanDDA analysis is performed in the PanDDA directory.

Suggested Workflow

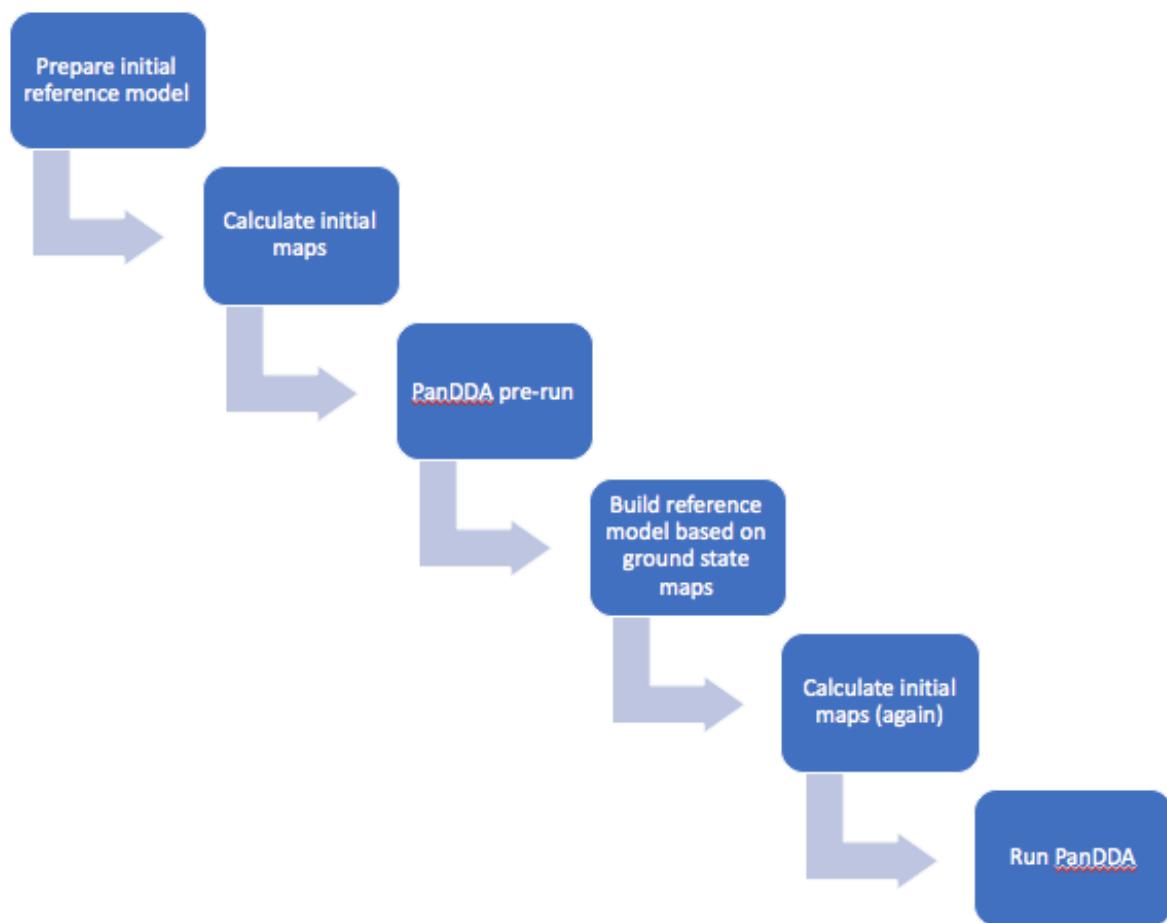


Figure 14. Step-by-step overview of the protocol required for creating a ground-state model for PanDDA analysis.

Ground-state model building

This protocol will help you to build a ground state model suitable for a X-ray fragment screening data analysis on XChem.

This method put the XChemExplorer data-management tool and on the PanDDA tool to good use. Basically, it consists in using either the XChemExplorer(XCE)-PanDDA data analysis outcome of a fragment pre-screen (100-150 fragments soaked) or a XCE-PanDDA pre-run outcome to build a

correct ground state model for the forthcoming PanDDA analysis of the full screen. It consists in six steps shown in the figure 1 above.

The ground state model represents the protein structure when no ligand is bound to it. However, it is not only a representation of the protein in its apo form. As it is calculated from data sets collected on hundreds of “ligand-free crystals” (with no ligand bound), it represents how the protein is in hundreds of ligand-free crystals. In addition, this ground state model will be used as reference model for building ligand. Updated models with ligand are “bound state” models which are a representation of the protein when a ligand is bound to it. What will be refined is the “ensemble model” which is a combination of the ground state model with the bound state model. This model explains the contribution to each structure factors in a more comprehensive way.

Therefore, modelling the right ground state model thus helps to capture and model rigorously every changed state due to a ligand binding. It also makes possible to rapidly have crystallographically correct protein-ligand models ready for computational chemistry analysis.

Initial reference model selection and quick preparation

Before you can enter the XCE workflow, you need to take a model from somewhere, *i.e.* a model straight from the PDB or from previous experiments and put it into the reference directory for calculation of initial maps. Make sure that the model is of reasonable quality and representative of the crystals used in the soaking campaign.

For example, it's always a good idea to take one of the best datasets from your solvent characterization experiment to build a good initial reference model.

A good initial reference model is a model with good stereochemistry, devoid of local errors, with good overall refinement statistics and build-in solvent molecules. However, don't overdo it at this stage, it's better to spend a bit more time later when you actually build the ground state model!

Calculating dimple maps and models in XCE

Once the initial reference model is ready, use this model to calculate electron density maps and models for all of the collected data sets using XCE (“maps” tab).

Here we are going to present you a workflow based on a PanDDA pre-run which we will not use all the available datasets but just a batch made of the first 100 data sets.

But do not wait to finish to collect all of the hundreds of data sets following a full fragment screen to build your ground state model! A good strategy is to use the data of the fragment pre-screen (100-150 fragments soaked) instead. You will then have your ground state model ready for the full PanDDA analysis which takes longer to run!

PanDDA pre-run

After the maps are calculated, in XCE, change to the PanDDA tab and select ‘pre-run for ground state model’ from the respective action box.

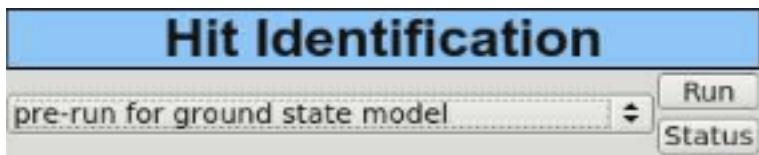


Figure 15. Action box for launching a PanDDA pre-run.

A dialog window will appear which prompts you to enter an appendix.

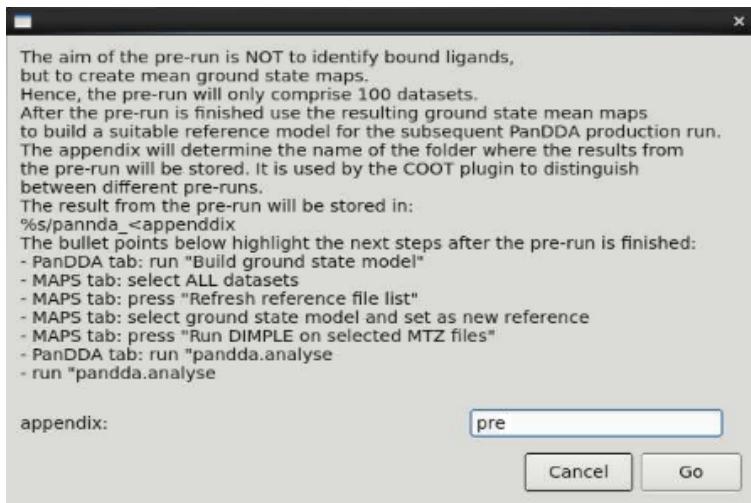


Figure 16. Dialog window with instructions of what to do after the pre-run is completed.

What will happen is: XCE will create a folder with the name *pandda_<appendix>* in the reference directory.

You can in principle launch several pre-runs, which may be useful if you are dealing with multiple crystal forms.

As the name implies, this is just a pre-run and the goal is not to identify ligands, but to get the ground state mean maps which will serve as a guide for building a proper ground state model for a later PanDDA production run. Because this is just a pre-run, this action will not use all the available datasets, but just the first 100 in order to save time.

Once PanDDA analyse is finished, XCE will automatically launch a script which will search for the most suitable reference mode amongst the 100 datasets used in the pre-run.

- It will parse the respective PanDDA logfile and look at all the datasets which belong highest resolution shell.
 - It will ignore all the datasets which contain interesting events since these have potentially ligands bound and are therefore not suitable as reference files.
 - From the remaining ones, it will take the one with the lowest Rfree and link the respective PDB & MTZ files as *<sampleID>-ground-state.pdb* and *<sampleID>-ground-state.free.mtz* into the reference folder.

It seems like a reasonable assumption to choose the ground state model according to these criteria, however, there is really no theoretical justification for doing so. If you think another model would be more suitable then simply link or copy the respective PDB & MTZ files as *<my_model>-ground-state.pdb* and *<my_model>-ground-state.free.mtz* file into the reference folder.

Make sure that the MTZ file contains columns labelled as F, SIGF and Rfree_flags.

Ground state model building

After PanDDA is finished, select '*Build ground state model*' from the respective action box.

XCE will not tell you when PanDDA is finished. The only way to find out is to check the respective PanDDA directory, in this case <reference_directory>/pandda_<appendix>. You know that PanDDA has finished once a file named pandda.done is present in the folder. If a file called pandda.errorred is present you may need to check the logfiles to find out what went wrong.

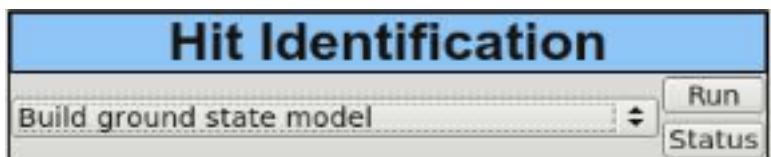


Figure 17. Action box for launching a COOT plugin which is specifically designed to build a ground state model.

This will launch a new COOT interface which allows you to modify and refine the initial reference model which was selected as described in first chapter.

Please note that no $2f_{\text{ofc}}$ and f_{ofc} maps will be displayed when you launch it the first time. These maps will only become available after the first cycle of refinement.

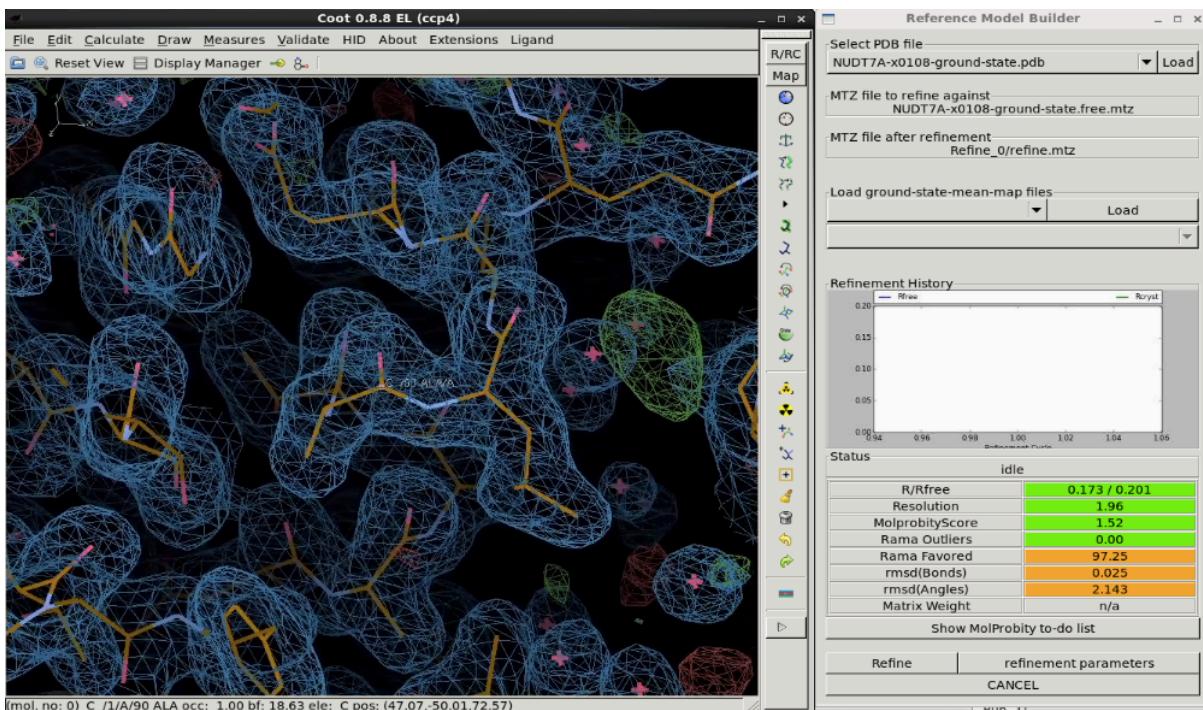


Figure 18. COOT interface for building and refinement of the ground state model.

You can now select the PanDDA pre-run from which you want to display the respective ground state mean maps. In the example given here, the folder name is panddatest.

This will take a while because it will load one ground state mean map for every resolution shell!

It will by default only display the one from the highest resolution shell, but you can use the combo box to select the respective maps from any resolution shell.

It is not yet clear if this is really a useful feature or if it just unnecessarily slows down the program; feedback is welcome!

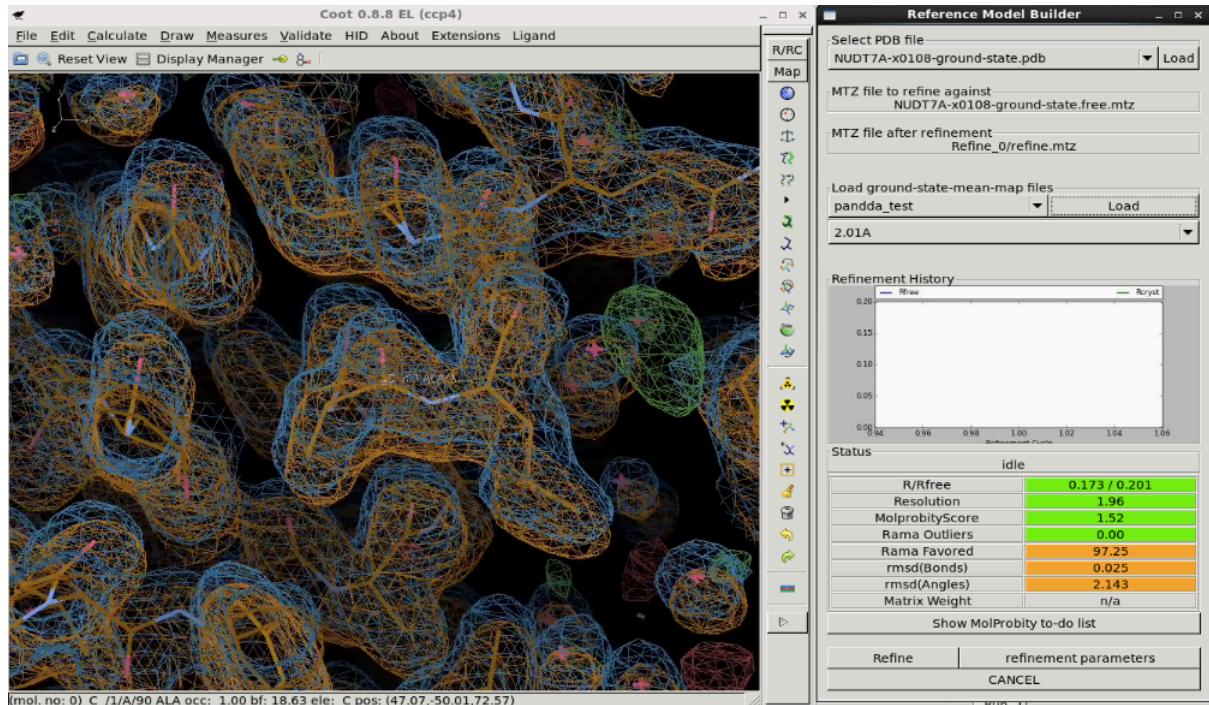


Figure 19. COOT interface with 2fofc maps (blue), fofc maps (green/blue) and ground state mean maps (brown).

You can now modify and refine the initial reference model so that it explains the ground state mean map.

Build or remove all side chains alternative conformation, all the water molecules, all the solvent molecules you have now evidence of. It is now worth spending a bit of time to build, as much as you can, a fairly accurate reference model which will then become your ground state model.

By doing so, you will save even more time in the forthcoming steps. First, you will save time during the PanDDA event inspection in pandda.inspect as less “blobs/event” due to missing features of the reference model will be found during the PanDDA analysis. Secondly, validating interesting models will also take you less time as all they all will be already well-refined after the first cycle of refinement.

Keep in mind that the mean maps are only shown so that you can understand what the mean ground state of all crystals looks like. Since this is not a crystallographic map, it does not contain any information that will be used during refinement!

Once you are happy with the first modifications of your model, adjust your refinement parameters and press ‘Refine’. The following bullet points describe what happens next:

- A new folder named <sampleID>ground-state will be created in the reference directory, if it does not already exist from previous refinements.
- Depending on the refinement cycle, a folder named Refine_<cycle> will be created in the <sampleID>ground-state folder and the modified PDB file from COOT will be saved as in.pdb.

- The modified model will be refined on the DLS cluster. A spinner will appear in the refinement panel and it will spin until the refinement is finished (see Figure 20). You cannot launch another refinement job while the current refinement is not finished.
- After the refinement is finished, the symbolic link named <sampleID>-ground-state.pdb will be removed from the reference directory and replaced by a link with the same name, but which points to the file that came out of refinement.
- In COOT, the old PDB file and the 2fofc and fofc maps will be deleted and the new PDB file and 2fofc & fofc maps which came out of refinement will be loaded. You can now iterate between model building and refinement until you have a model that fits your ground state maps and retains good stereochemistry, is devoid of local errors and has good overall quality statistics.

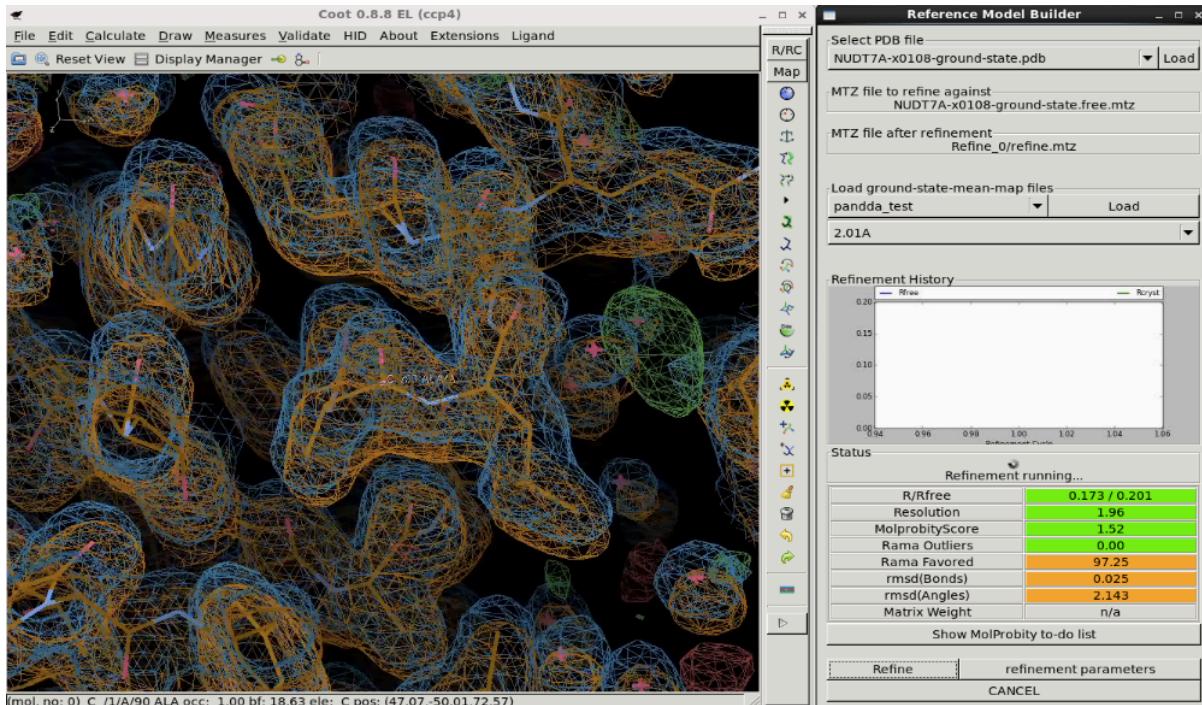


Figure 20. COOT interface during refinement. Note the little spinner in the middle of the panel which indicates that the refinement is still running.

Congratulation! You now have a correct ground state model/reference model for the PanDDA full analysis.

Map re-calculation

After you have finished building the ground state model, go back to the MAPS tab of your current instance of XCE.

We would recommend to first remove all the dimple files in order to avoid that any unwanted files remain! You can use the option in the "Maps and Restraints" action box.

The new ground state model is already in the reference folder, but XCE does at this point not know that it exists. First press 'Refresh reference file list', then select the respective ground state model and finally press 'Set New Reference (if applicable)'. Now you're good to go to calculate a new set of maps which can then be used for the PanDDA production run!

The screenshot shows the XChemExplorer software interface with the 'Maps' tab selected. The main area is a table with 21 rows of data, each representing a sample. The columns include Sample ID, Select, Compound ID, Smiles, Resolution, Dimple Rfree, Data Processing Group, Reference Space Group, Difference IC Volume (%), Reference File, Data Processing Unit Cell, Dimple Status, Compound Status, and Last. The last row is currently selected. Below the table are four buttons: 'Update Tables From Datasource', 'Datasets', 'Maps & Restraints', 'Hit Identification', and 'Refinement'. The 'Hit Identification' button is highlighted with a purple background.

Figure 21. XCE Maps tab for calculating a new set of maps with a new ground state model.

[pandda.analyse](#)

--- coming soon ---

[pandda.inspect](#)

--- coming soon ---

[PanDDA model export to the project directory](#)

--- coming soon ---

[Convert Event Maps to MTZ files](#)

You need to convert your Event Maps into MTZ format if you want to include them into your deposition or if you want to create an HTML summary file. In order to do so, go to the PANDDA tab and select 'Event Map -> SF' from the 'Hit identification' action box (Figure 22).



Figure 22. Event map conversion.

When you press 'RUN', XCE will search all the folders in the project directory (!) and convert every event map that it finds into MTZ format. The conversion runs on the local machine, so this can take quite a while if you collected many datasets and have many event maps. Note that it currently can only convert the maps into a P1 MTZ file.

Refinement

Refinement stages



Figure 23 Refinement progress model of XCE

The figure summarises the refinement stage model that XCE uses to track the progress of each sample and which is also used to triage samples. You need at least do some initial refinement to be able to look at samples in the refinement interface.

Overview

A list of all models that are currently ‘in refinement’ can be viewed in the ‘Refinement’ tab.

	Sample ID	Compound ID	Refinement Space Group	Refinement Resolution	Refinement Rcryst	Refinement Rfree	Refinement Outcome	PanDDA site details	Refinement Status												
1	NUDT22A-x0106	N13421a	P 21 21 21	1.39	0.20663	0.22948	3 - In Refinement	<table border="1"> <tr><td>Index</td><td>Name</td><td>Status</td></tr> <tr><td>2</td><td>Crystal contact</td><td>3 - In Refinement</td></tr> </table>	Index	Name	Status	2	Crystal contact	3 - In Refinement	finished						
Index	Name	Status																			
2	Crystal contact	3 - In Refinement																			
2	NUDT22A-x0161	N14124a	P 21 21 21	1.43	0.20613	0.23017	3 - In Refinement	<table border="1"> <tr><td>Index</td><td>Name</td><td>Status</td></tr> <tr><td>1</td><td>Mg site & putasteric site 1</td><td>4 - CompChem ready</td></tr> <tr><td>4</td><td>near xtal contact</td><td>3 - In Refinement</td></tr> </table>	Index	Name	Status	1	Mg site & putasteric site 1	4 - CompChem ready	4	near xtal contact	3 - In Refinement	finished			
Index	Name	Status																			
1	Mg site & putasteric site 1	4 - CompChem ready																			
4	near xtal contact	3 - In Refinement																			
3	NUDT22A-x0182	N14004a	P 21 21 21	1.48	0.21242	0.23591	3 - In Refinement	<table border="1"> <tr><td>Index</td><td>Name</td><td>Status</td></tr> <tr><td>1</td><td>Mg site & putasteric site 1</td><td>4 - CompChem ready</td></tr> </table>	Index	Name	Status	1	Mg site & putasteric site 1	4 - CompChem ready	finished						
Index	Name	Status																			
1	Mg site & putasteric site 1	4 - CompChem ready																			
4	NUDT22A-x0196	N14099a	P 21 21 21	1.75	0.21716	0.25772	3 - In Refinement	<table border="1"> <tr><td>Index</td><td>Name</td><td>Status</td></tr> <tr><td>1</td><td>Mg site & putasteric site 1</td><td>4 - CompChem ready</td></tr> </table>	Index	Name	Status	1	Mg site & putasteric site 1	4 - CompChem ready	finished						
Index	Name	Status																			
1	Mg site & putasteric site 1	4 - CompChem ready																			
5	NUDT22A-x0202	N13854a	P 21 21 21	1.40	0.21239	0.23580	3 - In Refinement	<table border="1"> <tr><td>Index</td><td>Name</td><td>Status</td></tr> <tr><td>1</td><td>Mg site & putasteric site 1</td><td>4 - CompChem ready</td></tr> </table>	Index	Name	Status	1	Mg site & putasteric site 1	4 - CompChem ready	finished						
Index	Name	Status																			
1	Mg site & putasteric site 1	4 - CompChem ready																			
6	NUDT22A-x0215	N13708a	P 21 21 21	1.56	0.21362	0.24442	3 - In Refinement	<table border="1"> <tr><td>Index</td><td>Name</td><td>Status</td></tr> <tr><td>1</td><td>Mg site & putasteric site 1</td><td>4 - CompChem ready</td></tr> <tr><td>2</td><td>Crystal contact</td><td>3 - In Refinement</td></tr> <tr><td>5</td><td>Met1 and other missing resl</td><td>3 - In Refinement</td></tr> </table>	Index	Name	Status	1	Mg site & putasteric site 1	4 - CompChem ready	2	Crystal contact	3 - In Refinement	5	Met1 and other missing resl	3 - In Refinement	finished
Index	Name	Status																			
1	Mg site & putasteric site 1	4 - CompChem ready																			
2	Crystal contact	3 - In Refinement																			
5	Met1 and other missing resl	3 - In Refinement																			

Figure 24 Refinement tab in XCE

If you want to inspect them in COOT and if necessary want to refine them further, choose ‘Open COOT’ from the magenta action box and press RUN. F



Figure 25 Refinement task box

This will launch COOT and the respective XCE interface (Figure 26).

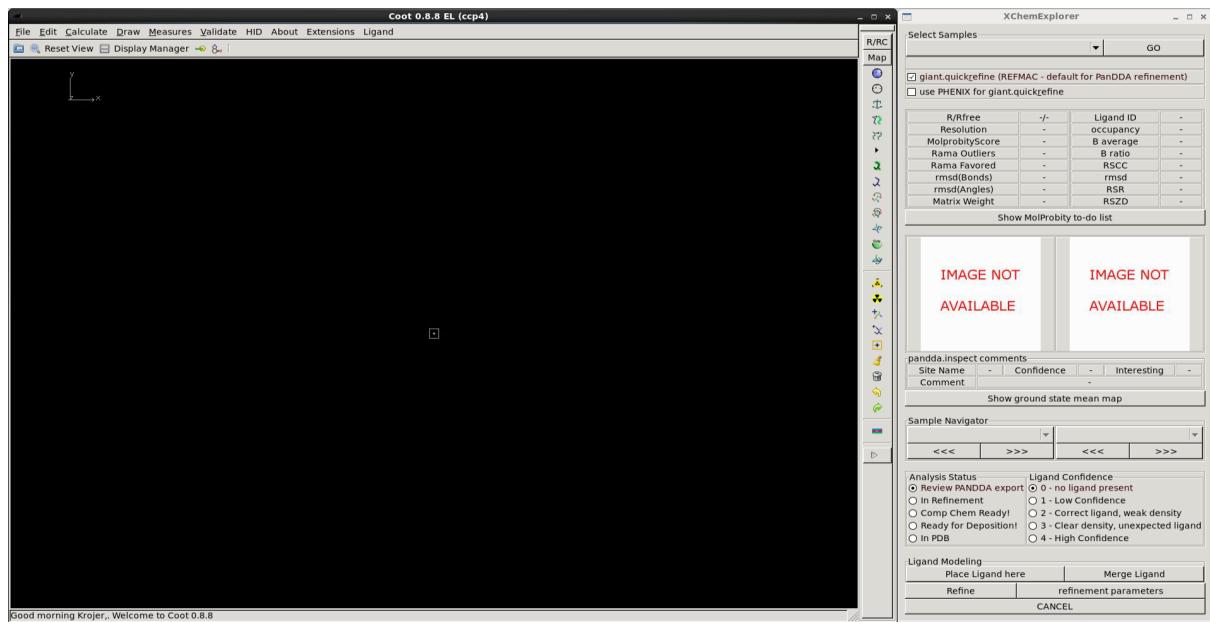


Figure 26 XCE Refinement interface for COOT

First you need to select the subset of samples you want to review/ refine from the drop-down menu at the top of the XCE interface (Figure 26). The drop-down lets you choose the Refinement Stage and once you press GO will load all samples that are in the respective Refinement stage. The image below shows you all the available categories

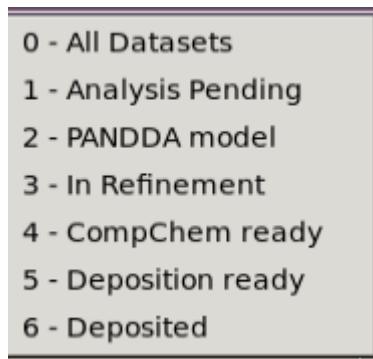


Figure 27 Sample selection criteria.

For example: if you want to load all models which are already in refinement, then select category 3 and press GO. In the example below, 79 structures are currently being refined (Figure 28).

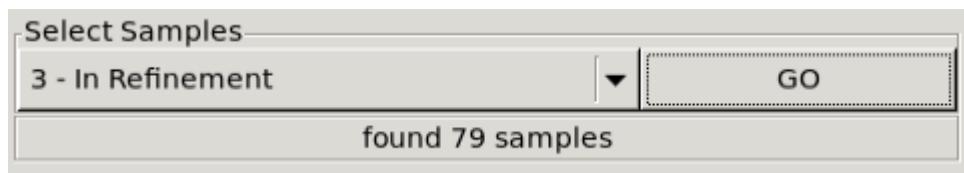
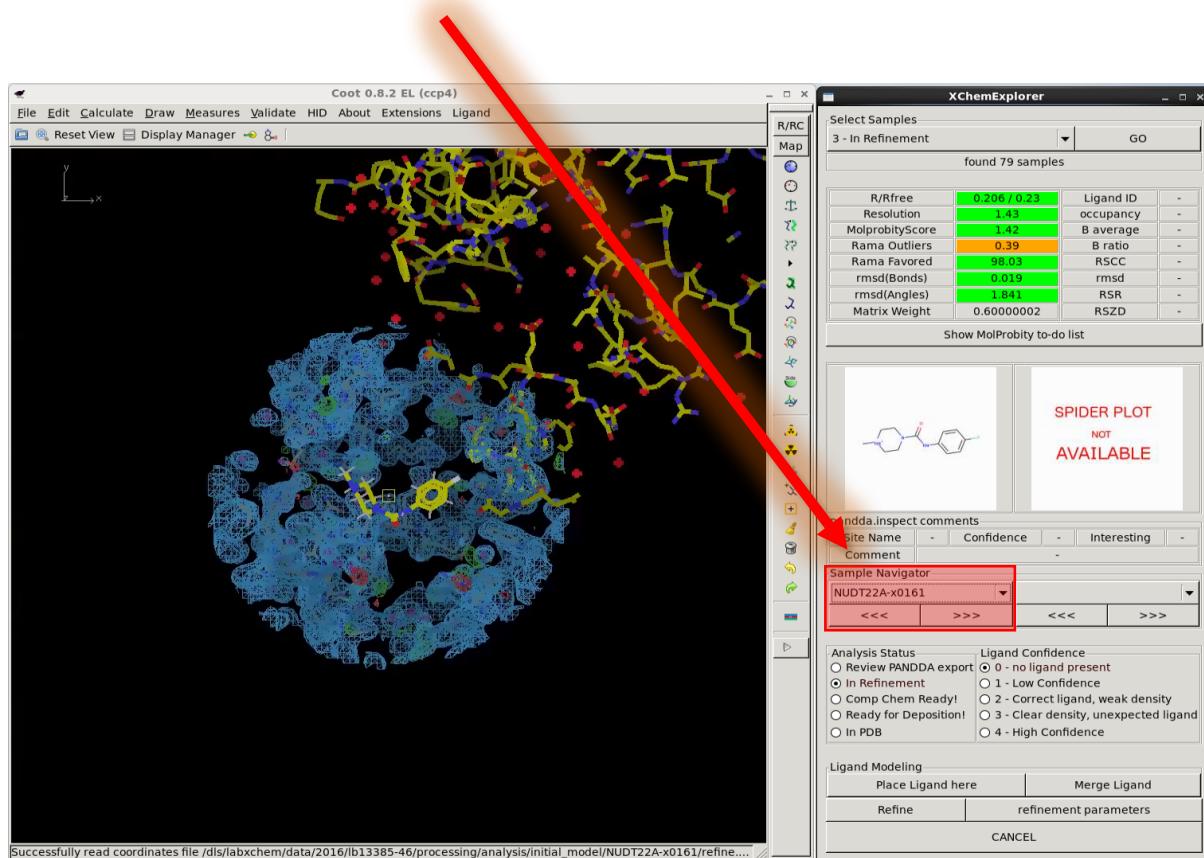


Figure 28 Exemplary response after sample selection.

In the left column of the ‘Sample Navigator’ Section, use the arrow buttons or the drop-down menu to select the structure of interest:



XCE will load the structure¹, 2fofc map (blue), fofc (green/red) map and a pdb file (+ dictionary) of the ligand if the latter was created and specified in the database. This ligand molecule may be slightly confusing because it may seem to just float in space. However, the molecule is completely

¹ Structure refers to the file called refine.pdb in the respective sample directory, or if no refinement has been carried out so far and category 0 or 1 are selected, then it will try to load dimple.pdb. Note: if you use XCE throughout the process, then refine.pdb as well as dimple.pdb are not actual files but symbolic links that point to the most recently refined file.

ignored as long as it is not merged into the main structure! It is only loaded to enable quick modelling if the ligand is not already part of the structure.

Deposition

XCE can be used to generate files for deposition via the new group deposition tool at the PDB (<https://deposit-group-1.rcsb.rutgers.edu/groupdeposit>) (Figure 29).

The screenshot shows the 'Group Deposition System' page. On the left, there's a sidebar titled 'Existing deposition' with fields for 'Group ID' and 'Password', and buttons for 'Log in' and 'Forgot Password'. On the right, a large blue header bar says 'Start a new deposition'. Below it, a message welcomes users to the system and provides instructions for starting a new deposition. A red note at the top of the main form area reads: 'Question about an in-progress deposition? For fastest response, login into your session and select the "Communication" page from the left hand navigation panel.' At the bottom of the form, there are fields for 'Your e-mail address', 'Password' (described as a shared 'group password' of 6 to 16 alphanumeric characters), 'Country', and 'Please copy this code : 29813'. A 'Start deposition' button is at the bottom right.

Figure 29. PDB Group Deposition website.

In order for a dataset to be considered for deposition, the *RefinementOutcome* flag has to be set to '5 - Deposition ready'. Once this is done, the respective sample will appear as a new line in the *depositTable* of the database.

Adding compulsory information

Every deposition needs information about title, authors, crystallization condition etc. This section describes how to enter this section. From the *Deposition* menu select 'Edit information' (Figure 30).

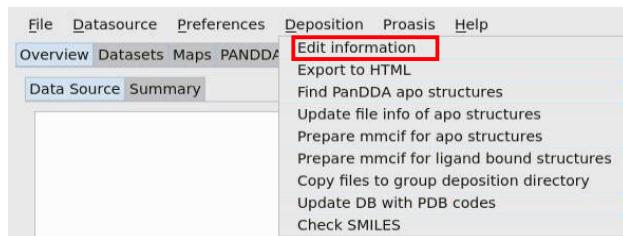


Figure 30. Deposition menu.

This will open a new window which provides a mask for input of all the essential pieces of information (Figure 31).

Note: you can use this mask to save identical information for ALL structures to be deposited.
However, this may not be suitable in cases where the information is different for certain samples.
In such cases, please use for example SQLiteBrowser to edit the relevant fields in the depositTable.

Figure 31. XCE entry widget for essential deposition information. Note that the compulsory fields have a white background whereas the fields where the user can provide additional information have a grey background.

It is possible to fill out all or part of the information and save the entered values into a file. Alternatively, it is possible to download an exemplary .deposit file from the XCE website (see Resources section in <http://tkrojer.github.io/XChemExplorer>). Download the file and select ‘Load File’ and the fields of the widget should change:

Note: you can use this mask to save identical information for ALL structures to be deposited.
However, this may not be suitable in cases where the information is different for certain samples.
In such cases, please use for example SQLiteBrowser to edit the relevant fields in the depositTable.

Figure 32. XCE entry widget for essential deposition information with data.

Most of the fields are self-explanatory, however, it is worth drawing the attention to the *General* tab where one can provide information about the title. The given example contains two wildcards, *\$ProteinName* and *\$CompoundName*. The \$ sign indicates that XCE will fill in whatever is given as ProteinName and CompoundName in the database. While the former will most likely be the same for all the entries so that one can provide the specific protein name instead of a wildcard, it is useful to provide the compound name one wants to see in the title of the PDB deposition in the database. Once all the information is entered, use ‘Save file’ to save a blueprint in case something needs changing, then press ‘Save to Database’ in order to update all the entries that are currently in the deposition table. Note that if you add more samples later, you will again have to press again ‘Save to Database’.

The screenshot shows the DB Browser for SQLite interface. The main window displays the 'depositTable' table with the following schema and data:

```

    CREATE TABLE depositTable (
        ID INTEGER,
        CrystalName TEXT,
        StructureType TEXT,
        PDB_file TEXT,
        MTZ_file TEXT,
        imCIF_model_file TEXT,
        mmcIF_SF_file TEXT,
        label TEXT,
        description TEXT,
        t_author_Pl TEXT,
        _author_Pl TEXT
    );

```

Data:

ID	CrystalName	StructureType	PDB_file	MTZ_file	imCIF_model_file	mmcIF_SF_file	label	description	t_author_Pl	_author_Pl	
1	NUDT7A-x0140	ligand_bound	NULL	NULL	NUDT7A-x014...	NUDT7A-x014...	NULL	NULL	Dr.	Frank	
2	NULL	NUDT7A-x0149	ligand_bound	NULL	NULL	NUDT7A-x014...	NUDT7A-x014...	NULL	NULL	Dr.	Frank
3	NULL	NUDT7A-x0151	ligand_bound	NULL	NULL	NUDT7A-x015...	NUDT7A-x015...	NULL	NULL	Dr.	Frank
4	NULL	NUDT7A-x0159	ligand_bound	NULL	NULL	NUDT7A-x015...	NUDT7A-x015...	NULL	NULL	Dr.	Frank
5	NULL	NUDT7A-x0220	ligand_bound	NULL	NULL	NUDT7A-x022...	NUDT7A-x022...	NULL	NULL	Dr.	Frank
6	NULL	NUDT7A-x0234	ligand_bound	NULL	NULL	NUDT7A-x025...	NUDT7A-x025...	NULL	NULL	Dr.	Frank
7	NULL	NUDT7A-x0286	ligand_bound	NULL	NULL	NUDT7A-x028...	NUDT7A-x028...	NULL	NULL	Dr.	Frank
8	NULL	NUDT7A-x0298	ligand_bound	NULL	NULL	NUDT7A-x029...	NUDT7A-x029...	NULL	NULL	Dr.	Frank
9	NULL	NUDT7A-x0303	ligand_bound	NULL	NULL	NUDT7A-x030...	NUDT7A-x030...	NULL	NULL	Dr.	Frank
10	NULL	NUDT7A-x0304	ligand_bound	NULL	NULL	NUDT7A-x030...	NUDT7A-x030...	NULL	NULL	Dr.	Frank
11	NULL	NUDT7A-x0374	ligand_bound	NULL	NULL	NUDT7A-x037...	NUDT7A-x037...	NULL	NULL	Dr.	Frank
12	NULL	NUDT7A-x0318	ligand_bound	NULL	NULL	NUDT7A-x031...	NUDT7A-x031...	NULL	NULL	Dr.	Frank
13	NULL	NUDT7A-x0434	ligand_bound	NULL	NULL	NUDT7A-x043...	NUDT7A-x043...	NULL	NULL	Dr.	Frank
14	NULL	NUDT7A-x0703	ligand_bound	NULL	NULL	NUDT7A-x070...	NUDT7A-x070...	NULL	NULL	Dr.	Frank
15	NULL	NUDT7A-x0740	ligand_bound	NULL	NULL	NUDT7A-x074...	NUDT7A-x074...	NULL	NULL	Dr.	Frank
16	NULL	NUDT7A-x1203	ligand_bound	NULL	NULL	NUDT7A-x120...	NUDT7A-x120...	NULL	NULL	Dr.	Frank
17	NULL	NUDT7A-x1205	ligand_bound	NULL	NULL	NUDT7A-x120...	NUDT7A-x120...	NULL	NULL	Dr.	Frank
18	NULL	NUDT7A-x1206	ligand_bound	NULL	NULL	NUDT7A-x120...	NUDT7A-x120...	NULL	NULL	Dr.	Frank
19	NULL	NUDT7A-x1210	ligand_bound	NULL	NULL	NUDT7A-x121...	NUDT7A-x121...	NULL	NULL	Dr.	Frank
20	NULL	NUDT7A-x1213	ligand_bound	NULL	NULL	NUDT7A-x121...	NUDT7A-x121...	NULL	NULL	Dr.	Frank
21	NULL	NUDT7A-x1235	ligand_bound	NULL	NULL	NUDT7A-x123...	NUDT7A-x123...	NULL	NULL	Dr.	Frank
22	NULL	NUDT7A-x1237	ligand_bound	NULL	NULL	NUDT7A-x123...	NUDT7A-x123...	NULL	NULL	Dr.	Frank
23	NULL	NUDT7A-x1244	ligand_bound	NULL	NULL	NUDT7A-x124...	NUDT7A-x124...	NULL	NULL	Dr.	Frank
24	NULL	NUDT7A-x1248	ligand_bound	NULL	NULL	NUDT7A-x124...	NUDT7A-x124...	NULL	NULL	Dr.	Frank

Figure 33. depositTable of the XCE database.

The interface obviously assumes that the information will be the same for every entry. But depending on the project, this may not be the case. Therefore, if you want to edit information for a specific entry, this is best done through the SQLite browser tool (Figure 1). Navigate to the *collectionTable* and then change the information as you see fit. Do not forget to press ‘*write changes*’ once you have finished! But remember, if you add more samples later and you press ‘*Save to Database*’, all the changed information will be overwritten. This is not ideal and will be addressed in one of the next software iterations.

Preparation of files for PDB upload

XCE will create a single gzipped tar file for direct upload into the Protein Data Bank. The *Group Deposition Directory* specifies the location where the file will be written to (see Settings, page 7). The next step is rather simple, from the *Deposition* menu choose ‘*Prepare mmcif for ligand bound structures*’ and XCE will start the prepare separate mmcif files for the PDB file and MTZ files for each entry in the deposition table. Note that if this deposition is part of a PanDDA analysis, then the program will also add Fourier coefficients for the event maps to the structure factor mmcif file. This may take a while depending of the number of files that are earmarked for deposition.

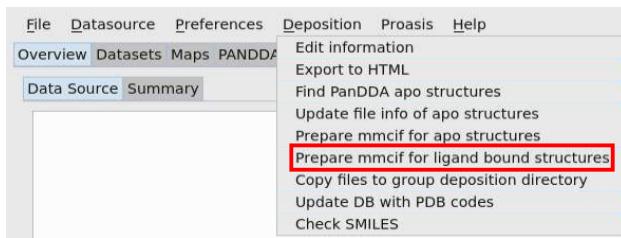


Figure 34. Prepare mmcif for ligand bound structures.

At the end, all mmcif files will be combined into a gzipped tar archive file, which can be directly uploaded into the PDB. At the moment, the only way to follow the progress of the process is by monitoring the output in the terminal window.

Troubleshooting

```
grep ERROR xce.log
```

Note: the XCE GUI does currently contain several obsolete items, which will be removed during the next software iteration.

Tips & Tricks

DLS remote connection

NX is a great way of working remotely at Diamond. However, it is not always ideal if you have to do extensive rebuilding in *pandda.inspect* or during refinement. The method described here uses *Filesystem in Userspace* (FUSE) to recreate the Diamond file system at your home institution. All you need is a reasonably fast internet connection and a friendly IT team:

First, your IT team (unless you use your private computer) needs to install FUSE (<https://github.com/libfuse/libfuse>):

e.g. **sudo yum install fuse-utils sshfs** (for Centos distributions)

or **sudo apt-get install sshfs** (for Ubuntu)

See <https://gist.github.com/cstroe/e83681e3510b43e3f618> for details (note that FUSE is also available for Mac). Avoid building from source unless you really have to.

Then, they need to create a /dls mount point:

e.g. **sudo mkdir /dls**

I am not sure if the next step is always necessary, but they had to change the ownership of the directory at SGC (Note xtalnmr is the group I belong to):

e.g. **sudo chown tkrojer:xtalnmr /dls**

Your IT team needs to set this up, but from now on, you can mount and unmount the Diamond file system whenever you want. You can mount the Diamond file system with the following command:

sshfs <fed_id>@nx.diamond.ac.uk:/dls /dls

e.g. **sshfs zqr16691@nx.diamond.ac.uk:/dls /dls**

And that's it. Now go into your labxchem visit and fire up XCE!

Note: if you cannot convince your IT team that this is a good idea, an alternative is to run linux through a virtual machine. Then you are ROOT and can do whatever you want!

Another thing to consider: FUSE seems to uphold the connection while you are working, but it seems to drop it if you are away from keyboard for a while, i.e. while going for lunch.

Now go to your labxchem directory, then launch XCE.

e.g. **cd /dls/labxchem/data/2017/1b18145-3/processing**

But before you start, here are some considerations of how to run XCE:

Option 1 is to install XCE locally. You can find download and installation instructions on GitHub:

<https://github.com/xchem/XChemExplorer>

Several people have done this and it seems to work well. The only dependency is that you need to have ccp4 v7.0 or higher installed on your system.

Option 2 is to launch the DLS version of XCE on your machine. We haven't tested this extensively at this point, but it would be the preferred mode because you will then not have to worry about updating XCE etc. In principle, all you need to do is run the following command:

`/dls/science/groups/i04-1/software/XChem/xce`

There may be still a few glitches, therefore, any feedback would be greatly appreciated.

Note: XCE and COOT now run on your local machine, but the files are pulled from Diamond and written back to the Diamond file system. Depending on the speed of your internet connection, it may take a bit to load big map files. However, you will see it's worth it!

And once you're done, unmount the file system:

`fusermount -u /dls`

Using XChemExplorer outside the labxchem environment

XCE was over the years optimised to support crystallographic fragment screening at the DLS XChem facility. It is however a generic tool and it will accept any data as long as they are organised in a certain hierarchical manner.

If you collect data at the Diamond Light Source, just make sure that you have set `${proteinname}/${samplename}` as your data collection directory and you are good to go. XCE is currently only able to parse the auto-processing results at DLS, but it should not be too difficult to adapt it to other synchrotron sites. Just get in touch in case you think this could be useful for your research!

The following section will describe what you need to do if your data are already processed and you want to import them into XCE.

Project Directory structure

XCE is essentially a file system based tool. It uses a SQLite database to record results and outcome and to track progress, but it can restore most of the information from the file system alone. Not everything of course, but enough to continue refining and depositing crystal structures. Therefore, as long as the data adhere to a few conventions and are present in the expected XCE directory structure, they can be imported. Please also check the

Database section for how to generate and edit a new database.

All files are stored in the same project directory and have to be organised in a hierarchical manner, i.e. files belonging to one crystal are stored in a subfolder and the name of this folder is equal to the sample identifier. The filenames have to adhere to a naming convention, but the requirements are minimal and it is easy to add data manually by adjusting folder and file names to the expected nomenclature. All that needs to be provided are an MTZ and an AIMLESS logfile in the respective sample directory. The files must have the same filename root as the folder name, e.g. if the folder name is TEST-x001, then the MTZ and logfile must be called TEST-x001.mtz and TEST-x001.log, respectively.

The following example will illustrate the data structure further:

The project directory structure of XChemExplorer is as follows:

<project_directory>/<sample_id>

e.g.

/Users/tobiaskrojer/SGC/PHIPA/fragment_screen/PHIPA-x001

/Users/tobiaskrojer/SGC/PHIPA/fragment_screen/PHIPA-x002

/Users/tobiaskrojer/SGC/PHIPA/fragment_screen/PHIPA-x003

etc.

Each sample folder needs to contain a MTZ file and the corresponding AIMLESS logfile, e.g.:

PHIPA-x001.mtz

PHIPA-x001.log

XCE uses a file called <sample_id>.free.mtz as input for refinement. This file is either automatically generated by the DIMPLE difference map pipeline or users can choose to append an existing Rfree set from a reference file by providing it in the reference folder. If you have already done some refinement and you have already an Rfree set in your MTZ file, then just make a symbolic link, e.g.

ln -s PHIPA-x001.mtz PHIPA-x001.free.mtz

MTZ column labels must have CCP4 default names, otherwise XCE may show unexpected behaviour, i.e. IMEAN, SIGIMEAN, F, SIGF, FreeR_flag. Additionally, the program can only parse AIMLESS logfiles at the moment.

After DIMPLE was run successfully, the resulting PDB and MTZ files will be linked as dimple.pdb and dimple.mtz into the respective sample directory.

Once the refinement stage is reached a subfolder for each refinement cycle will be created: Refine_<cycle number>. This subfolder contains the modified PDB file, executable shell script and output. The script contains the complete refinement and validation schedule. After successful refinement, the resulting PBD and MTZ files will be linked as refine.pdb and refine.mtz into the sample directory. Again, you can provide the files yourself!

It is possible to create this folder structure manually and then choose Data Source -> Update Data Source from filesystem from the XCE menu to import all the information into the database.

This is all you need to do. It does not matter if these are real files or symbolic links. Exemplary sample folders could look like this once you are finished:

XXX

Updating the database

The database will already contain most of the information after you ran ‘Update Datasource from filesystem’. But if you want to edit or add more information like compound IDs or smiles strings (or anything else) then refer to Database update (page 12).

Running PanDDA

--- coming soon ---

Refinement

--- coming soon ---

Deposition

This chapter explains how to prepare data for deposition in case this is the only task you want to use XCE for. You need to prepare your data as described before, most importantly, the final PDB and MTZ files need to be present as *refine.pdb* and *refine.mtz* in the respective sample directories.

The only things left to do before you can continue with depositing your data as described in the Deposition (page 40), is to set the *RefinementOutcome* field in the *mainTable* of the database to ‘5 - Deposition ready’. This can be achieved by either manipulating the database with SQLite browser or by looking at your data in the XCE COOT interface and setting the *RefinementOutcome* field there.

Frequently asked questions

What happens when you step through the models?

XCE will load a file called refine.pdb (or dimple.pdb in case no refinement has been carried out so far) from the sample directory and if available a pdb file of the ligand and the respective restraints. Additionally, 2fofc as well as fofc maps are loaded (or they are calculated on the fly from refine.mtz/dimple.mtz if the map files were for whatever reason no pre-calculated). Note that refine.pdb/mtz and dimple.pdb/mtz are therefore reserved file names. Hence, if you wanted to manipulate your model outside the XCE environment you can easily do so and XCE will read the manipulated model in as long as it's called refine.pdb and present in the respective sample directory. One thing to keep in mind though: XCE carries out the actual refinement in a subfolder called Refine_<cycle number> and only links the resulting pdb/mtz files as refine.pdb/mtz into the project directory. Every time a new refinement is launched, it will first delete the respective symbolic links. But it will also delete it if it is an actual file. So if you want to keep the original, better create a symbolic link called refine.mtz. When you go through the models, it will remove all currently loaded models and load the aforementioned files from the next sample directory.

What happens when I make changes to the model?

All changes that you make to the model called refine.pdb will be preserved if you launch refinement (see next point). They will however be lost if you go to the next dataset. XCE will currently not ask if you want to keep the changes, the pdb file will be deleted from the list of molecules in COOT and it will be lost forever! Also, be careful if you read in additional molecules with the same name, for example if you want to analyse something. COOT does not mind if molecules have the same name since every molecule that is read in gets a unique internal identifier. XCE however recognises molecules by filename it may get confused in case of duplicates.

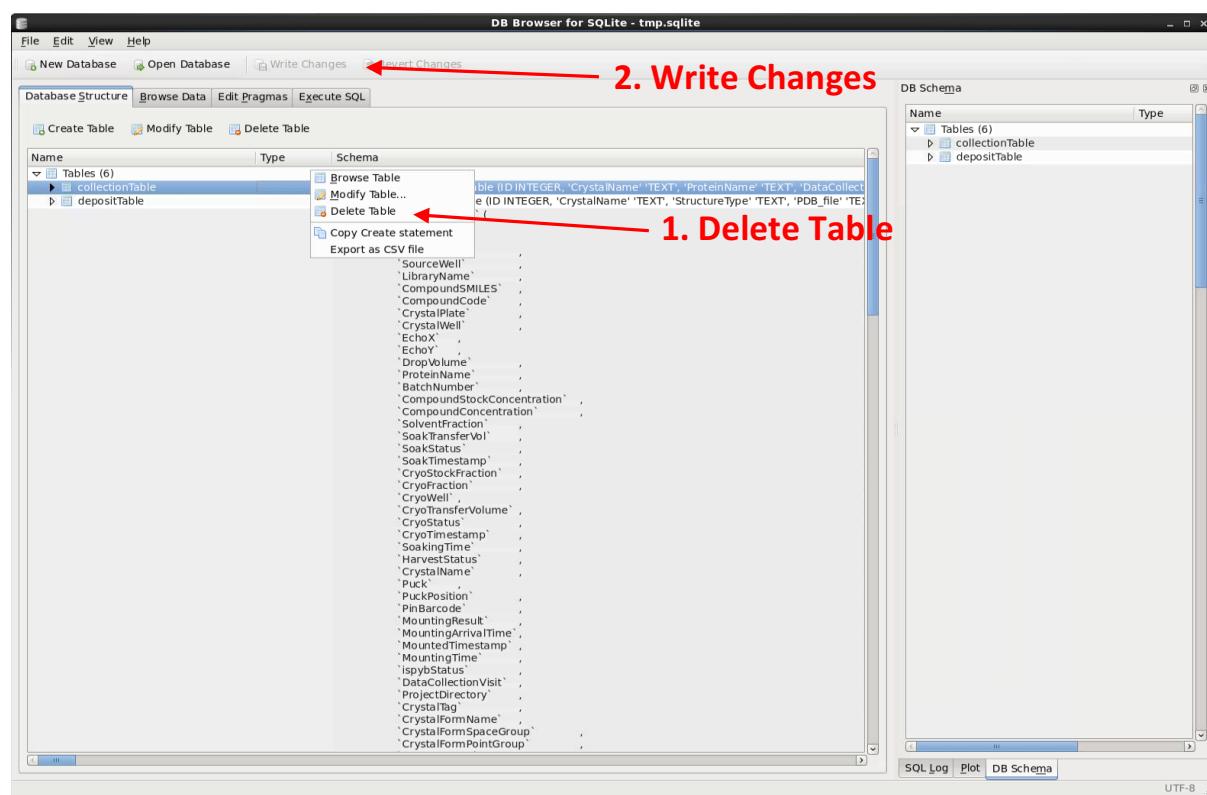
What happens when I refine the model?

When you press refine, XCE will take the model called refine.pdb and save it to a subfolder called Refine_<cycle number + 1> as in.pdb together with the shell script that will be used for refinement. If something goes wrong and the reason for failure is not clear it is usually a good starting point to look at the logfiles in the respective folder. Keep in mind that if you added for example water or other solvent molecules and did not merge them into refine.pdb, then they will not be included. XCE does not do any automatic merging!

I have wiped the contents of the project (initial_model) directory, but ‘Get new results from autoprocessing’ does nothing when I run it again

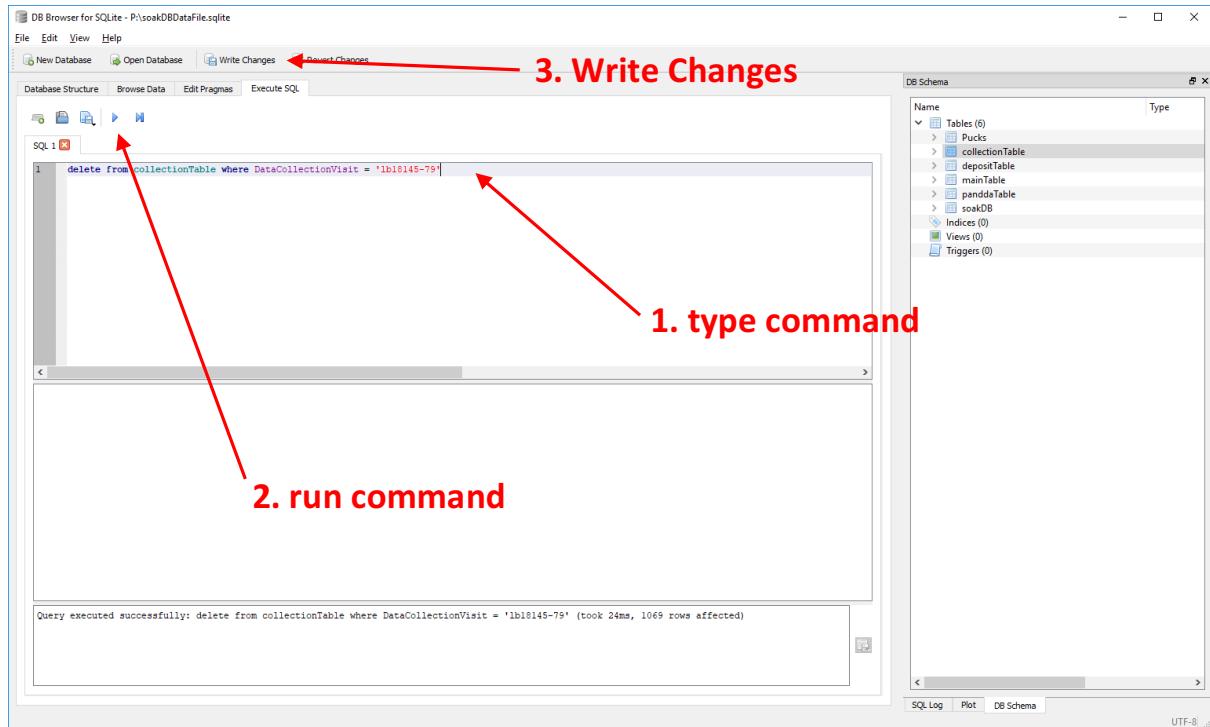
There may be instances when something gets messed up and a fresh start seems like the only option. Please note that if this is because XCE did something funny, let us know! You can delete the contents of your project directory at any time and as long as the data have not been backed up at Diamond, you will be able to recreate the folders. However, once you have run ‘Get new results from autoprocessing’ and all the files are copied over, XCE will not do anything after you have wiped the project directory. This is because XCE will first query the *collectionTable* in the database and check which autoprocessing results have already been copied over. It does not check if the actual files in the project directory exists, at this point. The only thing that matters for XCE is if an entry for a particular autoprocessing pipeline exists in the *collectionTable*. The reason why it works like this is because running an sqlite query is really fast, whereas checking if files exist is rather slow. So in order to copy the files over again, you can do two things:

Remove the complete collectionTable (if your project directory is empty)



Open your database file with SQLitebrowser. In the ‘Database Structure’ tab, right-click with the mouse on *collectionTable*; select ‘Delete Table’, then click ‘Write Changes’.

Remove all the autoprocessing results from a particular visit (if you only deleted folders/ crystals belonging to a certain data collection)



Open your database file with SQLitebrowser. Go to 'Execute SQL' tab. Then enter the command below; just change the visit (green, below) to whichever visit you want to delete:

```
delete from collectionTable where DataCollectionVisit = '1b18145-79'
```

Run the command and 'Write Changes'.

Known Problems

Bash vs C-shell

XCE was mostly tested in the bash shell. It should work in any of the C-shells, but there could be problems when XCE executes scripts in a subshell.