Xtrapol8 Command Line manual

November 24, 2021

This is a manual to use Xtrapol8 via command line.

Contents

1 Introduction

Xtrapol8 is software for the calculation of Bayesian-weighted Fourier difference (FoFo) maps, extrapolated structure factors and estimation of the occupancy. It is based on the cctbx toolbox [5] which come automatically with Phenix [6] and uses some CCP4 programs. [9] In order to run Xtrapol8, you will need to have a proper licence for Phenix and CCP4 and have both software suites installed. Please use Phenix 1.19 or higher.

Xtrapol8 can be used under Mac osx and Linux operating systems. It has not been tested on Windows but you are free to try.

2 Before you start

Clone or download the Fextrapolation repository to a place in your PATH or use the full path for running. Take care that all files are stored in the same directory.

Both CCP4 and Phenix should be setup via the command line. A more detailed description on how to setup CCP4 and phenix correctly for Xtrapol8 (Mac):

1. Open a terminal (can be found under *Utilities* or *Other* in *Launchpad*)
2. Add Phenix and the cctbx modules to your PATH:
   Source the file setpaths.sh withing the Phenix/build folder (you can use Finder to find out which Phenix version you have installed and where to find setpaths.sh).
   source /Applications/phenix-1.19.1-4122/build/setpaths.sh
3. Add CCP4 to your PATH:
   In the same terminal source ccp4.setup-sh from the ccp4/setup-scritps or ccp4/bin folder (again you can use Finder to find the file).
   source /Applications/ccp4-7.1/bin/ccp4.setup-sh
4. Check if Phenix and CCP4 programs can be found using the following commands. They should both return you the complete path.
   which phenix.refine
   which scaleit

You can add step 2 and 3 to your ∼/.profile, ∼/.zprofile or ∼/.bashrc file (this depends on your operation system and shell) if you want to avoid doing these steps before you run Xtrapol8 (if not then take care to do all steps in the same terminal window and run Xtrapol8 from the same terminal window).

3 Xtrapol8 organization

Fextr.py is the main script that calls the other scripts, and can be launched using an input file and/or command line arguments, in the same fashion as is done for phenix programs. Default values will be used for undefined parameters.
Processing consists of 5 steps:

1. Reading and checking input files, data quality assessment.
2. Calculation of (weighted) difference map.
3. Integrate in the difference maps and associate the peaks to the closest amino acid residues.
4. For each occupancy to test:
   - Calculate the requested extrapolated structure factors and map coefficients.
   - Quality assessment of the extrapolated structure factors.
   - Real space and reciprocal space refinement.
5. Estimate the occupancy of the triggered state based on the mFextr-DFc map or based on the structural comparison of the real space refined model with the starting structure.

We refer to the associated publication for more information. [3]

4 Running Xtrapol8 via command line

Fextr.py is the main script and should called with phenix.python.
phenix.python <where>[1]*/<to>/<find>/Fextr.py*

To avoid having to run the complete line, you can specify an alias in your ∼/.profile, ∼/.zprofile or ∼/.bashrc file:

[1] Words and numbers between < and > should be replaced by the actual words and values

alias X8='phenix.python <where>/<to>/<find>/Fextr.py'
Then run Xtrapol8 with
X8
instead of *phenix.python <where>/<to>/<find>/Fextr.py*

In what follows I will replace the <where>/<to>/<find>/Fextr.py just by Fextr.py, but remember that you should always provide the full path or the alias (*X8*) you defined above.

To get information on all options, use Xtrapol8 without any argument.
phenix.python Fextr.py

To run Xtrapol8, add an input file and/or add paramters via the command line.
phenix.python Fextr.py <input_file> <command_line options>

For some parameters, multiple input files can be added (e.g. input.additional_files). In that case, the keyword should be repeated. Not all parameters have to be specified, in that case the default value will be used. Take care that if a keyword is present in the input file or in the command line arguments, it needs to be specified. For some parameters "None" is accepted (see section 7).

Example using input file

1. Change the Xtrapol8.phil using your favorite editor
   nano Xtrapol8.phil

2. Run Xtrapol8
   phenix.python Fextr.py Xtrapol8.phil

Example using command line argument only

1. Run Xtrapol8 with all your arguments
   phenix.python Fextr.py input.reference_mtz=hiephiep.mtz input.triggered_mtz=hieperdepiep.mtz input.reference_pdb=hoera.pdb input.additional_files=jeej.cif input.additional_files=another.cif occupancies.list_occ=0.1,0.3,0.5 f_and_maps.f_extrapolated_and_maps=qfextr,qfgenick map_explorer.threshold=3.5 map_explorer.peak=4 output.outdir=fancy_party

Example using input file and command line

1. Change the Xtrapol8.phil using your favorite editor
   nano Xtrapol8.phil

2. Run Xtrapol8 with additional arguments. The order of arguments determines how parameters will be overwritten
   phenix.python Fextr.py Xtrapol8.phil refinement.phenix_keywords.refine.cycles=3

An example input .phil file with all options and an example with some minimal options can be found on the same location as where you saved the scripts. Another method to retrieve an input file is by directing the output of Xtrapol8 without arguments to a file:
phenix.python Fextr.py > Xtrapol8.phil

4.1 Input

- **reference_mtz**
  Data for the ground/untriggered state in mtz or mmcif format. For example, in case of a photo-induced process, this is the data when no laser light is applied; in case of a compound-induced reaction, this is the data before addition or release of the compound.

- **triggered_mtz**
  Data for the excited/triggered/perturbed state in mtz or mmcif format. For example, this is the data when the crystal(s) have been subjected to laser-light, a compound,... This crystals giving rise to this data usually occupy the triggered state in a small fraction (low occupancy).

Ideally, the unit cell and space group of this data is identical to the reference_mtz. If this is not the case, the space group and unit cell of the reference_pdb will be transferred to this data set. But then the results have to interpreted with great care and Riso and CCiso values will indicate whether or not this was allowed.

- **reference_pdb**
  Model for the ground/untriggered state in pdb or mmcif format. Preferably, this model has been refined with reference_mtz. Xtrapol8 has not yet been tested with a model containing nucleic acids.

- **additional_files**
  Additional library file for non-standard residues, in cif-format. Multiple cif-files can be added upon repetition of the keyword.

- **high_resolution**
  If no high resolution cutoff is specified, the highest possible resolution will be used. In practise, this is determined by the common reflections between reference_mtz and triggered_mtz.

- **low_resolution**
  If no low resolution cutoff is specified, the lowest possible resolution will be used. In practise, this is determined by the common reflections between reference_mtz and triggered_mtz. We do not recommend the use of a low resolution cutoff but it might be useful in specific cases.

If multiple suitable columns are found in the data files, then the following order will be applied for the reference data set:

1. If there are columns containing anomalous and non-anomalous signal, then columns containing the non-anomalous data will get priority over anomalous columns (but the Bijvoet pairs will be averaged and merged during processing as the treatment of anomalous data is not yet fully covered.)

2. If there are columns containing structure factors, then the structure factors will be directly used. If there are multiple columns with structure factors, then the first encountered columns containing structure factors and sigmas will be used.

3. If there are no columns with structure factors, then intensities will be used and converted to structure factors using truncate using the French-Wilson based scaling in truncate (CCP4).

The following order will be applied in order to extract data from the triggered data set:

1. Knowing the column labels of those columns that were selected from the reference data set, the same columns will be searched in the triggered data set.

2. If the same columns cannot be found, then the same search order as for the reference data set will be applied.

4.2 Occupancies

You need to specify a range of occupancies to test. This can be done in two ways:

- **low_occ**, **high_occ** and **steps**
  define the lowest and highest occupancies to test and the number of steps between them
  phenix.python Fextr.py <input file> low_occ=0.1 high_occ=0.3 steps=6

- **list_occ**
  provide the specific occupancies to test. If a list is provided, it will overwrite the previous method to define the occupancies.
  phenix.python Fextr.py <input file> list_occ=0.1, 0.25, 0.3

4.3 Scaling

The b_scaling option is used to specify the scaling method of the triggered_mtz versus the reference_mtz dataset:

- **anisotropic**
  Scale the triggered_mtz to the reference_mtz using an anisotropic B-factor scaling scheme.

- **no**
  Don't scale the two datasets. This option should only be used in case the two datasets are already scaled, e.g. when crystFEL is used for the data processing and the data are scaled and merged using partialator with custom-split option; or when XSCALE is used and the reference data set is added as reference.

- **isotropic**
  Scale the triggered_mtz to the reference_mtz using an isotropic B-factor scaling scheme.

4.4 Weighting of FoFo and extrapolated structure factors

- **q-weighting**
  Q-weighting uses Bayesian statistics to reduce the contribution of uncertain structure factor values to the $|F_{trig}| - |F_{ref}|$ difference structure factors. More information can be found in Ursby and Bourgeois, 1997. [8] Q-weighting can be applied for the calculation of $|F_{trig}| - |F_{ref}|$ difference structure factors as well as for extrapolated structure factors.

- **k-weighting**
  K-weighting is another weighting scheme to reduce the contribution of uncertain structure factor values to the $|F_{trig}| - |F_{ref}|$ difference structure factors introduced by Ren *et al.* [7] K-weighting can be applied for the calculation of $|F_{trig}| - |F_{ref}|$ difference structure factors as well as for extrapolated structure factors.

4.5 FoFo types

Using the possibility of q-weigting, k-weighting and no weighting, three possible $|F_{trig}| - |F_{ref}|$ difference structure factors can be calculated (Table 1):

- **qfofo**
  Q-weighted $|F_{trig}| - |F_{ref}|$ difference structure factors.

- **fofo**
  Non-weighted $|F_{trig}| - |F_{ref}|$ difference structure factors.

- **kfofo**
  K-weighted $|F_{trig}| - |F_{ref}|$ difference structure factors.
  K-weighting scale parameter allows to be more or less stringent about outlier rejection (f_and_maps.kweight_scale from 1.0 to 0.0, respectively).

Table 1: Three types of $|F_{trig}| - |F_{ref}|$ difference structure factors

type	Difference structure factors	Map Coefficients
qfofo	$q(F_o - F_o) = \frac{q}{<q>} \times (F_{trig} - F_{ref})$	$mq(F_o - F_o), \phi_{ref}$
fofo	$F_o - F_o = F_{trig} - F_{ref}$	$m(F_o - F_o), \phi_{ref}$
kfofo	$k(F_o - F_o) = \frac{k}{<k>} \times (F_{trig} - F_{ref})$	$mk(F_o - F_o), \phi_{ref}$

Only a single type of FoFo map can be calculated per Xtrapol8 run.

4.6 Extrapolated structure factors and map coefficients

Three different methods for calculation of extrapolated structure factors can be used. With the additional possibility to weight the structure factor difference, gives this rise to nine possibilities (Table 2):

- **qfextr, kfextr and fextr**
  As described in Coquelle et al., 2018. [1]

- **qfgenick, kfgenick and fgenick**
  As described in Genick et al., 2007. [4]

- **qfextr_calc, kfextr_calc and fextr_calc**
  As described in Schmidt (?)[cite]

Multiple maps and types can be selected in one run.
phenix.python Fextr.py <input file> f_and_maps.f_extrapolated_and_maps=fgenick
phenix.python Fextr.py <input file> f_and_maps.f_extrapolated_and_maps=qfextr,fextr,qfgenick

The keywords **all_maps**, **only_qweight** and **only_no_weight** allow an easy filterng over the structure factors and map to calculate.
To calculate all the nine map types:
phenix.python Fextr.py <input file> f_and_maps.all_maps=True
To calculate the three q-weighted map types:
phenix.python Fextr.py <input file> f_and_maps.only_qweight=True
To calculate the three k-weighted map types:
phenix.python Fextr.py <input file> f_and_maps.only_kweight=True
To calculate the three non-weighted map types:
phenix.python Fextr.py <input file> f_and_maps.only_no_weight=True

Take care that you can only select one of these four filtering options, and upon combination the settings will be overwritten in the following order: all_maps > only_qweight > only_kweight > only_no_weight.

Table 2: Nine types of Extrapolated structure factors and map coefficients

type	**Extrapolated structure factors**	**Map Coefficients**	
		2Fo-Fc type	**Fo-Fc type**
qfextr	$qF_{extr} = \alpha \times \frac{q}{<q>} \times (F_{trig} - F_{ref}) + F_{ref}$	$2m\|qF_{extr}\| - D\|F_c\|, \phi_{ref}$	$m\|qF_{extr}\| - D\|F_c\|, \phi_{ref}$
kfextr	$kF_{extr} = \alpha \times \frac{k}{<k>} \times (F_{trig} - F_{ref}) + F_{ref}$	$2m\|kF_{extr}\| - D\|F_c\|, \phi_{ref}$	$m\|kF_{extr}\| - D\|F_c\|, \phi_{ref}$
fextr	$F_{extr} = \alpha \times (F_{trig} - F_{ref}) + F_{ref}$	$2m\|F_{extr}\| - D\|F_c\|, \phi_{ref}$	$m\|F_{extr}\| - D\|F_c\|, \phi_{ref}$
qfgenick	$qF_{gen} = \alpha \times \frac{q}{<q>} \times (F_{trig} - F_{ref}) + F_{ref}$	$m_{dark}\|qF_{gen}\|, \phi_{ref}$	$m_{dark}\|qF_{gen}\| - D\|F_c\|, \phi_{ref}$
kfgenick	$kF_{gen} = \alpha \times \frac{k}{<k>} \times (F_{trig} - F_{ref}) + F_{ref}$	$m_{dark}\|kF_{gen}\|, \phi_{ref}$	$m_{dark}\|kF_{gen}\| - D\|F_c\|, \phi_{ref}$
fgenick	$F_{extr} = \alpha \times (F_{trig} - F_{ref}) + F_{ref}$	$m_{dark}\|F_{gen}\|, \phi_{ref}$	$m_{dark}\|F_{gen}\| - D\|F_c\|, \phi_{ref}$
qfextr_calc	$qF_{extr\_calc} = \alpha \times \frac{q}{<q>} \times (F_{trig} - F_{ref}) + F_{calc}$	$2m\|qF_{extr\_calc}\| - D\|F_c\|, \phi_{ref}$	$m\|qF_{extr\_calc}\| - DF_c\|, \phi_{ref}$
kfextr_calc	$kF_{extr\_calc} = \alpha \times \frac{k}{<k>} \times (F_{trig} - F_{ref}) + F_{calc}$	$2m\|kF_{extr\_calc}\| - D\|F_c\|, \phi_{ref}$	$m\|kF_{extr\_calc}\| - DF_c\|, \phi_{ref}$
fextr_calc	$F_{extr\_calc} = \alpha \times (F_{trig} - F_{ref}) + F_{calc}$	$m\|F_{extr\_calc}\| - D\|F_c\|, \phi_{ref}$	$m\|F_{extr\_calc}\| - D\|F_c\|, \phi_{ref}$

$\alpha = \frac{1}{occupany}$

4.7 Fast and furious versus calm and curious

Xtrapol8 has two modes of operation: *fast and furious* and *calm and curious*. As the name implies, the first mode is a fast mode, ideal for a first run to estimate the parameters for a next run or to get a fast result. In fast and furious mode, the default parameters will be used for several parameters, thus it can also be seen as an unsupervised mode. The reason for being much faster than the calm and curious mode is that reciprocal and real space refinement will only be run using the extrapolated structure factors and maps of the for the estimated occupancy instead of running the refinements with all extrapolated structure factors, and this only for the qFextr map type. Fixed parameters are:

- fofo_type = qfofo
- f_extrapolated_and_maps = qfextr
- all_maps = False
- only_qweight = False
- only_kweight = False
- only_no_weight = False
- negative_and_missing = truncate_and_fill

- use_occupancy_from_distance_analysis = False

The default is to run Xtrapol8 in calm and curious mode. To run Xtrapo8 in fast and furious mode, use *phenix.python Fextr.py <input file> f_and_maps.fast_and_furious=True*

4.8 Map explorer

Map explorer concerns the analysis of the $|F_{trig}| - |F_{ref}|$ difference map and Fo-Fc map types (Table 2 last column) in order to estimate the occupancy of the triggered state. It needs different parameters:

- **peak**
  Cutoff to find positive and negative peaks in the difference map, in r.m.s.d.
- **threshold**
  Cutoff for blob integration (blob = peak from with more than 2 voxels).
- **radius**
  Maximal radius for blob annotation to closest amino acid residue. If not defined, the high resolution cutoff (defined or implied by the mtz files) will be used.
- **z_score**
  Z-score cutoff to select only only the highest integrated peaks.

The integrated peaks around the selected residues will be used to estimate the occupancy of the triggered state and annotate the most probable extrapolated structure factors. This is done by integration of the $|F_{trig}| - |F_{ref}|$ difference map and Fo-Fc map types calculated from the extrapolated structure factors (Table 2 last column). To do this occupancy estimation, a residue list has to be provided. This can be the residue list calculated with all $|F_{trig}| - |F_{ref}|$ peaks or the residuelist with only the highest peaks as defined by the z-score. At a certain point, the script will pauze for 5 minutes and wait for a residue list. You can provide one of the two residue lists that are automatically generated (residlist.txt or residlist_Zscore<z-score>.txt, see section 5) or one that has been manually changed. If no list is provided, the list with the highest peaks will be used (residlist_Zscore<z-score>.txt).

- **use_occupancy_from_distance_analysis**
  In case of running Xtrapol8 in calm and curious mode, a second method is applied to estimate the occupancy. This method is based on a structural comparison of reference_pdb with the real space refined models after reciprocal space refinement (<outname>_occ<occupancy>_<extrapolated structure factor type>-DFc_reciprocal_space_real_space.pdb, see section 5). This method can consequently be very dependent on the refinement parameters used. Even though the analysis is done and results stored, the estimated occupancy is not used unless the *use_occupancy_from_distance_analysis* is set to *True*
  phenix.python Fextr.py <input file> map_explorer.use_occupancy_from_distance_analysis=True
  This can be usefull in cases where the $|F_{trig}| - |F_{ref}|$ does not show high and pronounced peaks. Also for this occupancy estimation method a residue list has to be provided, in the same format as for the difference map comparison (actually the same will be used for both). This will be the residue list arriving from the largest peaks (residlist_Zscore<z-score>.txt) by default. All atoms from all residues will be compared if an empty list is provided. This can be computationally heavy and can take a very long time.

4.9 Refinement

At the point of refinement, 3 different refinements will subsequently be run:

1. Reciprocal reciprocal space refinement using the extrapolated structure factors and reference_pdb.
2. Real space refinement using the extrapolated map coefficients and reference_pdb.
3. Real space refinement using the map coefficients and model originating from the reciprocal space refinement (refinement 1).

- **run_refinement**
  Setting this parameter to False avoids doing any type of refinement. This is useful for any case in which a manual intervention is needed before refinement. This will be the case for ligand binding studies as the ligand is not present in the input model, when the conformational changes are too large to be captured by the automatic refinement or a bond break takes place. The automatic refinements and analyses can be still be executed using the Refiner.py script.

- **use_refmac_instead_of_phenix**
  By default the reciprocal and real space refinement will use phenix.refine and phenix.real_space_refinement, respectively. It is also possible to use refmac for the reciprocal space refinement and Coot for the real space refinement.
  phenix.python Fextr.py <input file> refinement.use_refmac_instead_of_phenix=True
  Take care that Refmac and Coot are in your PATH (which should be the case if ccp4 has been correctly sourced).

- **phenix_keywords**
  Some keywords for refinement by phenix can be changed using these parameters. We refer to the phenix documentation for more information. Density modification is performed using the program dm (CCP4). [2]

- **refmac_keywords**
  Some keywords for refinement by refmac can be changed using these parameters. We refer to the refmac documentation for more information. Density modification is performed using the program dm (CCP4). [2]

In fast and furious mode, refinement will only be performed using the extrapolated structure factors and map coefficients for the estimated occupancy and this only for qFextr extrapolation (see 4.7). In calm and curious mode, the three refinement steps will be performed for each occupancy estimation and for all of the requested extrapolated structure factor types.

The refinements will be performed using all the data between the high and low resolution cutoff as defined by the Xtrapol8 input parameters. For extrapolated structure factors, a different resolution cutoff might be more appropriate. This cutoff will be different for each type of extrapolated structure factors and occupancy. Even though a high resolution cutoff will be suggested in the generated output, it is the user's responsibility to check the data and decide on the actual application of such a resolution cutoff.

4.10 Output

- **outdir**
  Directory in which the output will be stored. If not specified, then the current directory will be used. If the output directory does not already exists, it will be created. If the directory already exists, Xtrapol8 will create a new one using the specified outdir name followed by a number.

- **outname**
  Name to be used as suffix or prefix in several output files. The name of the triggered_mtz will be used in case outname is not specified.

- **generate_phil_only**
  Create the input file and exit. This is a useful feature from when the GUI is used to create an input file that can be run in command line later, e.g. on a more powerful machine, or to merge command line parameters into the input file.

- **generate_fofo_only**
  Stop Xtrapol8 after the calculation of the Fourier difference map.

- **open_coot**
  Open automatically a COOT session with the $|F_{trig}| - |F_{ref}|$, extrapolated and refined density maps for the estimated occupancy, the input model and the refined models for the estimated occupancy. For each extrapolated structure factor type a single COOT script is written and stored in the folder of its best occupancy. When generating the FoFo only, then the session is stored in the output directory. The COOT script can also be launched afterwards using

coot - - script <path/to/Xtrapol8/output/directory/subdirectory>/coot_all_<extrapolated structure factor type>.py

- **ddm_scale**
  Scale factor for the ddm plot (see See 5.1). The ddm colors will range from -scale to +scale.

5 Output

The output directory contains all output files and subdirectories:

- **<date-and-time>_Xtrapol8.log**
  Log-file with most important output, such as a repetition of the input parameters, data quality statistics in table form, location of output files and occupancy determination.
- **Xtrapol8_in.phil** and **Xtrapol8_out.phil**
  For easy rerun and checking of the parameters.
  Whereas the input phil-file should be a copy of the input file and additional command line parameters, the output file takes changes that were made during the program into account.
- The $|\mathbf{F_{trig}}| - |\mathbf{F_{ref}}|$ **difference map coefficients** in three formats: mtz, ccp4 and xplor.
  The xplor-file is used for the map explorer analysis and can be removed afterwards to reduce disk space.
  See Tabel 3 for column names in the mtz-file.
- The output files from map explorer:
  - **peakintegration.txt**
    Contains the summary of the $|F_{trig}| - |F_{ref}|$ difference map peaks analysis (from cell explorer).
  - **residlist.txt**
    Contains all residues that have at least one associated difference map peak.
  - **residlist_Zscore<z-score>.txt**
    Contains all residues that have at least one of the highest associated difference map peak. If the peaks are not normally distributed, no z-score can be calculated and residlist_Zscore<z-score>.txt will be equal to residlist.txt.
- Figures
  See 5.1
- A subdirectory per occupancy that is tested:
  - Subdirectories called **qweight_occupancy_<occupancy>**
    Contain all output from q-weighted extrapolated structure factors for a specific occupancy.
  - Subdirectories called **kweight_occupancy_<occupancy>**
    Contain all output from k-weighted extrapolated structure factors for a specific occupancy.
  - Subdirectories called **occupancy_<occupancy>**
    Contain all output from non-weighted extrapolated structure factors for a specific occupancy.

  See 5.2
- **pymol_movie.py**
  Script to open all maps and models in Pymol with models as different frames from highest to lowest occupancy.
  Can be opened with Pymol (if installed)
  Pymol pymol_movie.py
  or from a pymol window:
  run <path/to/Xtrapol8/output/directory/>pymol_movie.py

5.1 Figures

New figures to be added

- **Riso_CCiso.pdf**
  Plot showing Riso and CCiso of reference_mtz and triggered_mtz versus resolution (Figure 1).
  An overall Riso value below 10 % indicates that the two data sets are isomorphous. However, this is often not the case which can deteriorate the quality of the maps and final results. An Riso value below 0.25 and CCiso above 0.75 for the highest resolution shell are highly recommended.

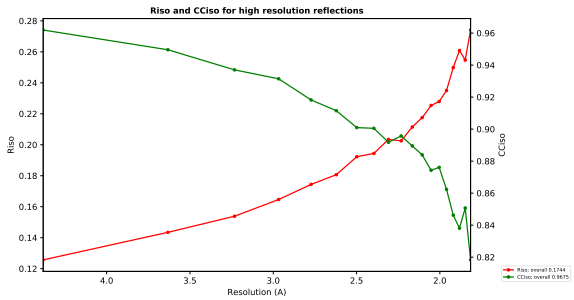

Figure 1: Example of plot showing the average Riso and CCiso values versus resolution. In this example, a high resolution cutoff at 1.9Å is advised.

- **Q_estimation.pdf**
  Plot showing the average q values and its range versus resolution. (Figure 2)

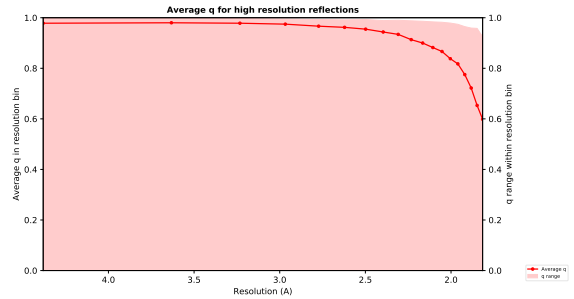

Figure 2: Example of plot showing average q value and its range versus resolution. In this example, the weight of the heigh resolution reflections is reduced as compared to the low resolution reflections.

- **k_estimation.pdf**
  Plot showing the average k-values and its range versus resolution. (Figure 3).

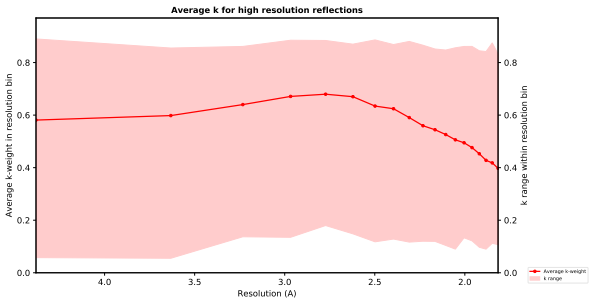

Figure 3: Example of plot showing average k value and its range versus resolution.

- **summed_difference_peaks.pdf**
  Plot showing the positive and negative integrated peak area versus amino acid and secondary structure (Figure 4). α-helices are depicted as pink bars whereas β sheets are depicted by blue triangels; positive

and negative peaks in the $|F_{trig}| - |F_{ref}|$ difference map are depicted as green and red bars, respectively. Ligands and water molecules, if they have associated difference map peaks as determined by map-explorer, are show on a separate plot at the bottom. For the ease of comparison, the y-axis has the same range in all subplots.

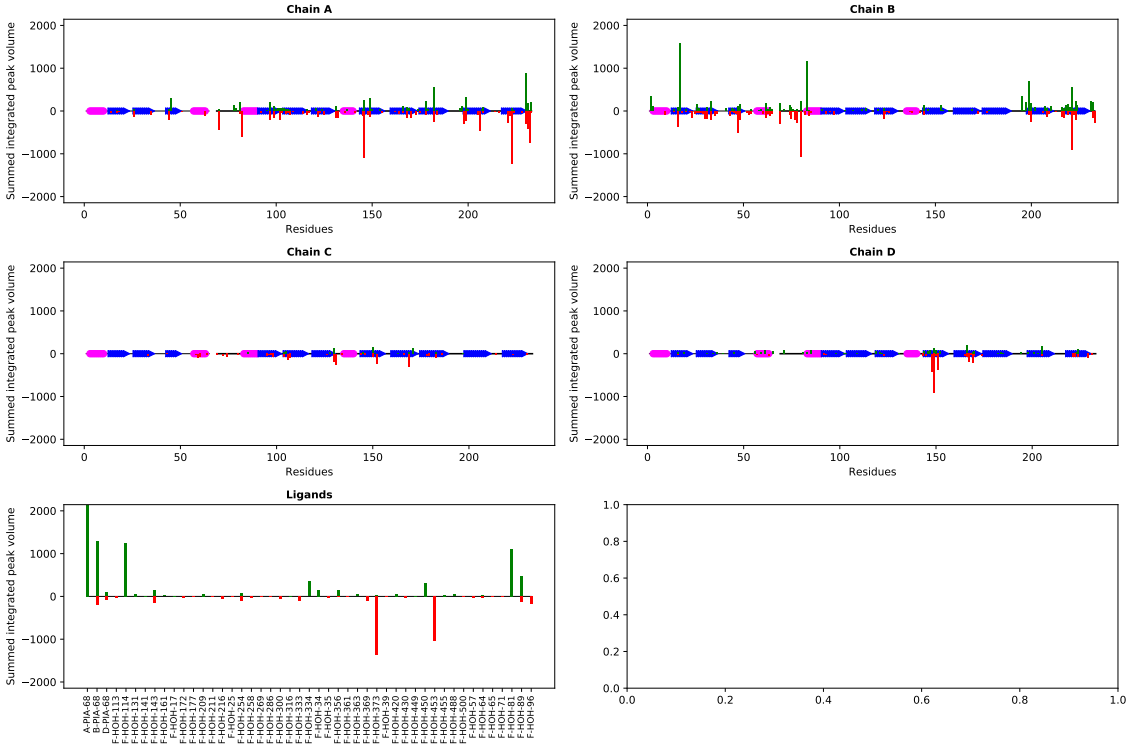

Figure 4: Example of secondary structure plot indicating the $|F_{trig}| - |F_{ref}|$ difference map peaks. The example model has four chains and several ligands and water molecules to which difference map peaks are annotated.

- **<q/k>FoFo_sigmas.pdf**
  Plot showing the the average $|F_{trig}| - |F_{ref}|$ values (full lines and spheres for data points, red, left axis) and their error estimation (dashed line with crosses for data points, blue, right axis) versus resolution (Figure 5).

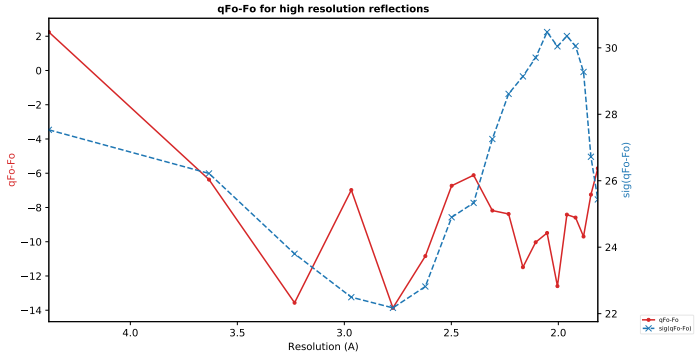

Figure 5: Example of plot showing the average $|F_{trig}| - |F_{ref}|$ structure factors and estimated errors. In this example, there are more negative differences than positive differences in each high resolution shell. The drop in sigma values at high resolution indicates that the sigma values are no longer correctly estimated

- **<extrapolated structure factor type>_sigmas.pdf**
  Plot showing the average extrapolated structure factors value (red to salmon) and their error estimation (blue to light blue) versus resolution. This plot is generated for each type of requested extrapolated

structure factors (Figure 6).

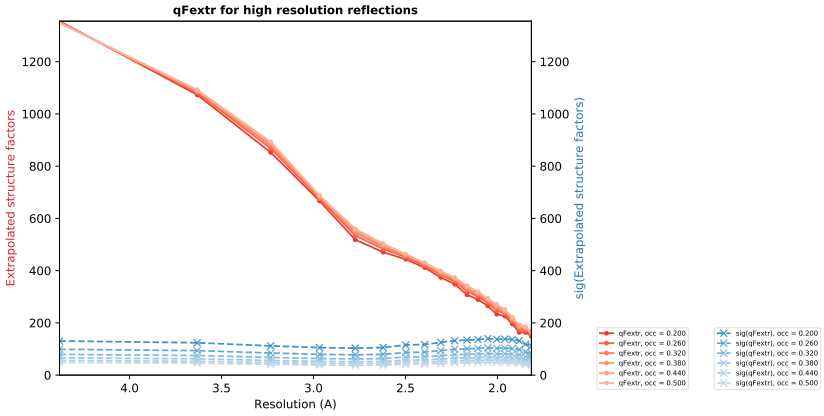

Figure 6: Pot showing the extrapolated structure factors and estimated errors for each type of extrapolated structure factors and occupancy. The bottom plot shows that the data strength decreases fast but at all resolution shells remain higher than the error values.

- **Neg_Pos_reflections_<extrapolated structure factor type>.pdf**
  Plot showing the absolute number (spheres, left axis) and percentage (crosses, right axis) of negative reflections in the extrapolated structure factors versus occupancy. This plot is generated for each type of requested extrapolated structure factors (Figure 7).

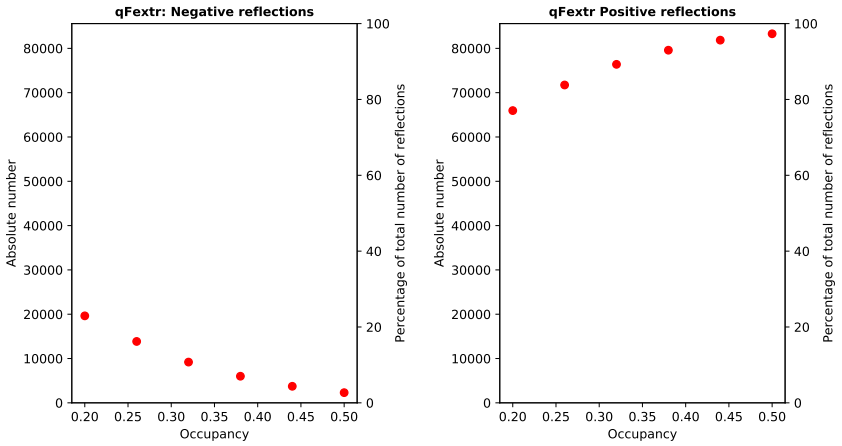

Figure 7: Example of plot showing the number of negative reflections for each extrapolated structure factor type versus the occupancy.

- **alpha_occupancy_determination_<extrapolated structure factor type>.pdf**
  Plot showing the normalised comparison between the difference maps to estimate alpha and occupancy, for the selected residues (from the residue list), all residues and the selected ones with an increased signal-to-noise ratio (Figure 8). The α/occupancy with a normalized ratio of one, is the estimated correct alpha/occupancy.
  A different picture is generated for each type of requested extrapolated structure factors.

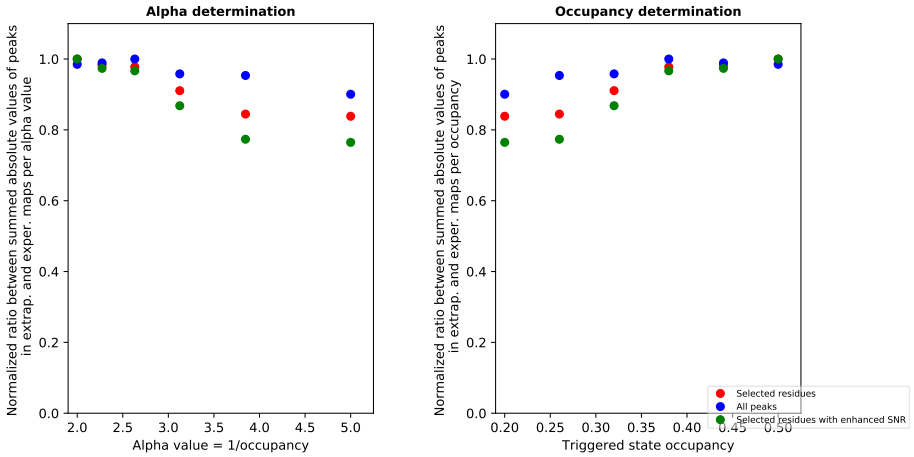

Figure 8: Example of plot showing the normalized difference map signal versus occupancy and alpha value for a single extrapolated structure factor type. In this example, there are only two significant distances. The fit of the average and average of the fits, superpose.

- **Distance_difference_plot_<extrapolated structure factor type>.pdf**
  Plot showing the distance differences between reference_pdb and the real space refined models (after reciprocal space refinement first) versus α.
  Briefly, all interatomic distances between the residue of the refined and reference_pdb are calculated for the residues in the list and plotted versus α. If an exponential fit can be made, they are maintained as significant. If less than 40 distances are significant, they are all plotted, otherwise they are not. Further the average of all significant distances is plotted and an exponential curve fitted. The alpha is estimated based on the average of the fits of all distances at 95% of reaching the plateau level, but alternative estimations are based on the fit of the average distance.
  This plot is only generated upon running Xtrapol8 in *calm and curious* mode because it is based on the real space refined models.
  A different picture is generated for each type of requested extrapolated structure factors.

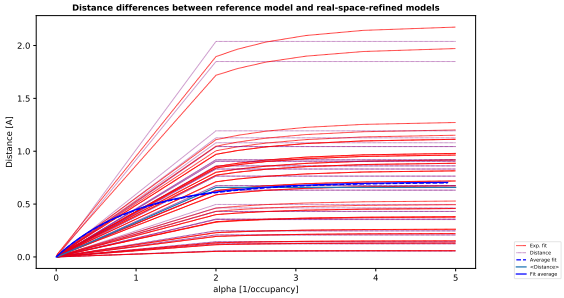

Figure 9: Example of plot showing the distances between atoms in the real-space refined models with those of model_in versus α value for a single extrapolated structure factor type.

- **<extrapolated structure factor type>_refinement_R-factors_per_alpha.pdf**
  Plot showing the Rwork, Rfree and Rwork-Rfree gap after reciprocal space refinement.
  This plot is only generated upon running Xtrapol8 in *calm and curious* mode because it is based on the reciprocal space refinement run for each occupancy.
  A different picture is generated for each type of requested extrapolated structure factors.

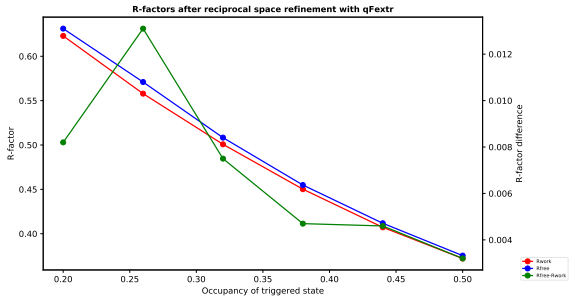

Figure 10: Example of plot showing the refinement R-factors after reciprocal space refinement verses alpha value for a single extrapolated structure factor type.

5.2 Subdirectory per occupancy value

Each subdirectory from the output directory contains the output specific to a tested occupancy and whether or not q-weighting is applied:

- <**outname**>_**occ**<**occupancy**><**extrapolated structure factor type**>**.mtz**
  The extrapolated structures factors. These can be used as input data for further refinement.
  See Tabel 3 for column names in the mtz-file.

- <**extrapolated structure factor type**>_**peakintegration.txt**
  Contains the summary of the peaks in the extrapolated difference map.

- <**extrapolated structure factor type**>_**residlist.txt**
  Contains all residues that have at least one associated difference map peak.

- Extrapolated map coefficients

  - <**outname**>_**occ**<**occupancy**>_<**extrapolated structure factor type**>**-DFc.mtz**
    2Fo-Fc and Fo-Fc type map coefficients associated to a certain map type in mtz-format.
    See Table 2 for calculation of the map coefficients. See Tabel 3 for column names in the mtz-file.
  - <**outname**>_**occ**<**occupancy**>_**2**<**extrapolated structure factor type**>**-DFc.ccp4**
    2Fo-Fc type map coefficients associated to a certain map type in ccp4-format.
    See Table 2 for calculation of the map coefficients.
  - <**outname**>_**occ**<**occupancy**>_<**extrapolated structure factor type**>**-DFc.ccp4**
    Fo-Fc type map coefficients associated to a certain map type in ccp4-format.
    See Table 2 for calculation of the map coefficients.
  - <**outname**>_**occ**<**occupancy**>_<**extrapolated structure factor type**>**-DFc.map**
    Fo-Fc type map coefficients associated to a certain map type in xplor-format.
    See Table 2 for calculation of the map coefficients.
    This file that is used for the map explorer analysis and can be removed afterwards.

- Figures
  Three figures are generated for each type of requested extrapolated structure factors. The first two figures gives an overview of the data quality of the extrapolated structure factors and can help to decide on a resolution cutoff for further refinement and usage.

  - <**extrapolated structure factor type**>_**negative_reflections.pdf**
    Plot showing the absolute number (red, left axis) and percentage (blue, right axis) of negative reflections in the extrapolated structure factors versus resolution, and plot showing the completeness versus resolution before data was manipulated to eliminating negatives(Figure 11). "True completeness" is the completeness of the positive reflections only. A completeness/True completeness above 90% is highly recommended, and will be different depending on the requested extrapolated structure factor type and occupancy.

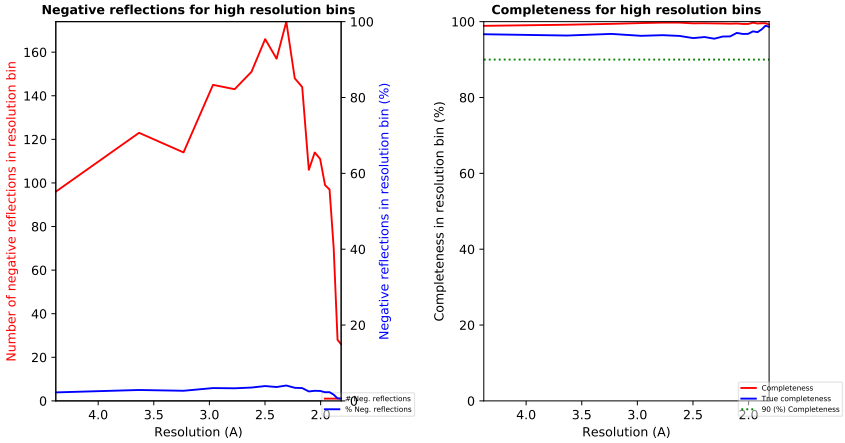

Figure 11: Example of plot showing negative reflections versus resolution on the right, and plot showing the completeness and true completeness on the right. The 90% completeness estimation is incorrectly estimated because the completeness is higher than 90% in all cases.

– **<extrapolated structure factor type>_occupancy<occupancy>_FsigF.pdf**
Plot showing the average extrapolated structure factors over the estimated error versus resolution (Figure 12).$\langle F/\sigma(F)\rangle$ above 1.22 is highly recommended ($\sim \langle I/\sigma(I)\rangle$ >1.5).

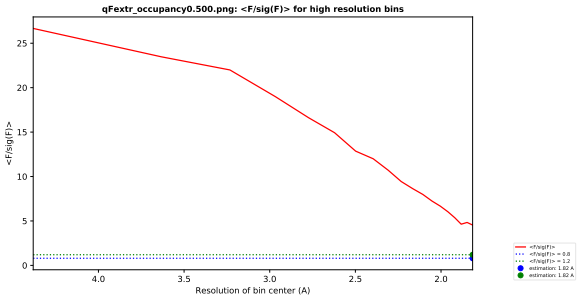

Figure 12: Example of plot showing $\langle F/\sigma(F)\rangle$ versus resolution. In this example, the resolution cutoff is badly estimated because the signal-to-noise ratio is high in each resolution shell.

– **ddm_<reference_pdb>_<reciprocal_space_real_space_refined_model>.pdf**
Plot showing the atomic differences between the reference_pdb and the real space refined model. This plot is only generated for the models with estimated occupancy.

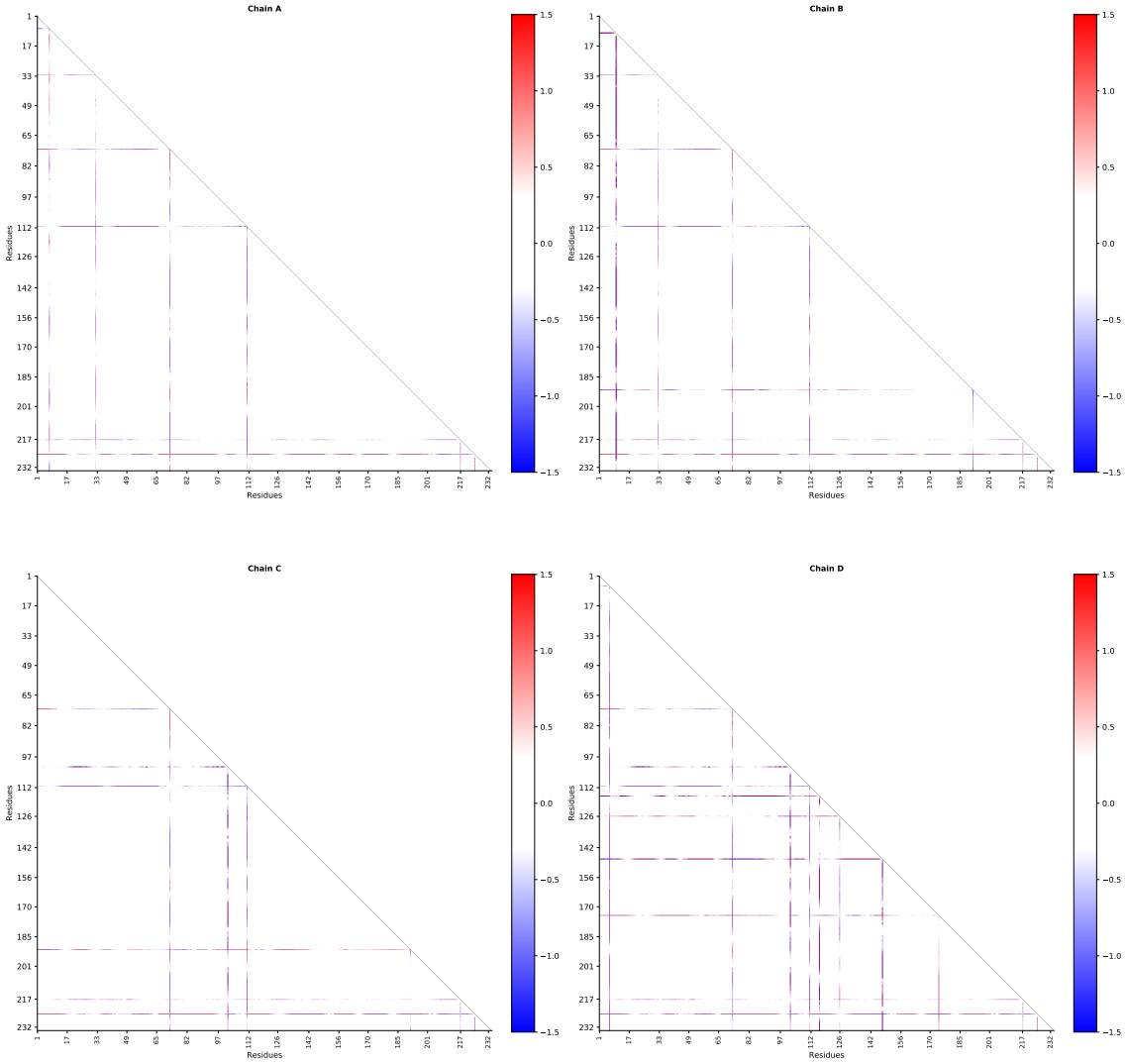

Figure 13: Example of plot showing a distance difference matrix (ddm) indicating for each protein chain the main atomic differences between the input model and the model (reciprocal space + real space refined) at the estimated occupancy.

- Output of refinements
  In case of running Xtrapol8 in *fast and furious* mode, this is only present for the folder of the estimated occupancy, otherwise this is present in all subdirectories.

  - <**outname**>_**occ**<**occupancy**>_<**structure factor type**>**-DFc_reciprocal_space.mtz** or <**outname**>_**occ**<**occupancy**>_<**structure factor type**>**-DFc_refmac.mtz** and <**outname**>_**occ**<**occupancy**>_<**structure factor type**>**-DFc_reciprocal_space.pdb** or <**outname**>_**occ**<**occupancy**>_<**structure factor type**>**-DFc_refmac.pdb**
    Output mtz and pdb file from reciprocal space refinement with phenix.refine or refmac, respectively, using the extrapolated structure factors and reference_pdb.
  - <**outname**>_**occ**<**occupancy**>_<**structure factor type**>**-DFc_real_space.pdb** or <**outname**>_**occ**<**occupancy**>_<**structure factor type**>**-DFc_coot_real_space_refined.pdb**
    Output pdb file from real space refinement with phenix.real_space_refine or Coot, respectively.
  - <**outname**>_**occ**<**occupancy**>_<**structure factor type**>
    -DFc_reciprocal_space_real_space.pdb or
    <**outname**>_**occ**<**occupancy**>_<**structure factor type**>

Table 3: mtz column labels

type	Difference structure factors		Map Coefficients	
	amplitudes	**sigmas**	**amplitues**	**phases**
qfofo	QFDIFF	SIGQFDIFF	QFOFOWT	PHIQFOFOWT
fofo	FDIFF	SIGFDIFF	FOFOWT	PHIFOFOWT
kfofo	KFDIFF	SIGKFDIFF	KFOFOWT	PHIKFOFOWT
type	**Extrapolated structure factors**		**Map Coefficients***	
	amplitudes	**sigmas**	**amplitues**	**phases**
qfextr	QFEXTR	SIGQFEXTR	2QFEXTRFCWT	PHI2QFEXTRFCWT
kfextr	KFEXTR	SIGKFEXTR	2KFEXTRFCWT	PHI2KFEXTRFCWT
fextr	FEXTR	SIGFEXTR	2FEXTRFCWT	PHI2FEXTRFCWT
qfgenick	QFGENICK	SIGQFGENICK	QFGENICKWT	PHIQFGENICKWT
kfgenick	KFGENICK	SIGKFGENICK	KFGENICKWT	PHIKFGENICKWT
fgenick	FGENICK	SIGFGENICK	FGENICKWT	PHIFGENICKWT
qfextr_calc	QFEXTR_CALC	SIGQFEXTR_CALC	2QFEXTR_CALCFCWT	PHI2QFEXTR_CALCFCWT
qfextr_calc	KFEXTR_CALC	SIGKFEXTR_CALC	2KFEXTR_CALCFCWT	PHI2KFEXTR_CALCFCWT
fextr_calc	FEXTR_CALC	SIGFEXTR_CALC	2FEXTR_CALCFCWT	PHI2FEXTR_CALCFCWT

* "FCWT" is added to the column label for the difference map of type Fo-Fc, .

-DFc_refmac_coot_real_space_refined.pdb
Output pdb file from real space refinement with phenix.real_space_refine or Coot after reciprocal space refinement with phenix.refine or refmac, respectively. These pdb-files are used in the distance analysis and ddm-plot, unless the reciprocal space refinement was unsuccesful (in the latter case the pdb files from the simple reciprocal or real-space refinement are used).

- **coot_all.py**
  Script to open all files in Coot.
  coot - -script coot_all_<extrapolated structure factor type>.py
  This is only present for the folder with the estimated occupancy.
  If the Coot executable is found in the PATH and output.open_coot=True, then this Coot session is automatically opened at the end of the program.

6 Trouble shooting

- *TypeError: coercing to Unicode: need string or buffer, NoneType found*
  One of the obligatory input files (reference_mtz, triggered_mtz or reference_pdb) is not defined or not defined correctly.
  Check your input file and/or command line arguments.
- *File not found: None*
  One of the obligatory input files (reference_mtz, triggered_mtz or reference_pdb) is not defined or not defined correctly.
  Check your input file and/or command line arguments.
- Problems with Refmac, statistic calculation, etc.
  Check if the resolution cutoff is properly chosen.
- *OSError: [Errno 2] No such file or directory:* There is a problem with the output directory.
  If you are running Xtrapol8 from the command line, please make sure that params.output.GUI equals False.
  Make sure you have the permission to create the output directory (make sure you have write permission).
- The output has overwritten a former Xtrapol8 run.
  If you are running Xtrapol8 from the command line, please make sure that params.output.GUI equals False.

7 List of available keywords

input
.reference_mtz = None
Reference data in mtz or mmcif format (merged).
.triggered_mtz = None
Triggered data in mtz or mmcif format (merged).
.reference_pdb = None
Reference coordinates in pdb or mmcif format.
.additional_files = None
Additional files required for refinement, e.g ligand cif file, restraints file.
.high_resolution = None
High resolution cutoff (Angstrom). Will only be used if high resolution of the input data files extends to this value.
.low_resolution = None
Low resolution cutoff (Angstrom).

occupancies
.low_occ = 0.1
Lowest occupancy to test (fractional)
.high_occ = 0.5
Highest occupancy to test (fractional)
.steps = 3
Amount of equaly spaced occupancies to be tested
.list_occ = None
List of occupancies to test (fractional). Will overwrite low_occ, high_occ and steps if defined

scaling
.b_scaling = no isotropic *anisotropic
B-factor scaling for scaling triggered data vs reference data.Cannot be used for reference data with fcalc when using mmtbx.fmodel.manager.

f_and_maps
.fofo_type = *qfofo fofo kfofo
Calculate q-weighted or non-q-weighted Fo-Fo difference map.Q-weighted is highly recommended.
.kweight_scale = 0.05
scale factor for structure factor difference in k-weigting scheme (for calculation of kfofo)
.f_extrapolated_and_maps = *qfextr fextr qfgenick fgenick qfextr_calc fextr_calc
qFextr, Fextr: (q-weighted)-Fextr structure factors and maps by Coquelle method (|Fextr|= α×(|Fobs,triggered|-|Fobs,reference|)+|Fobs,triggered|, map: 2m|Fextr|-D|Fcalc|, ϕ_model).
qfgenick, fgenick: (q-weighted)-Fextr structure factors and maps by Genick method (|Fextr|= α×(|Fobs,triggered|-|Fobs,reference|)+|Fobs,triggered|, map: m|Fextr|, ϕ_model).
qFextr_calc, Fextr_calc: (q-weighted)-Fextr structure factors and maps by Fcalc method (|Fextr|= α×(|Fobs,triggered|-|Fobs,reference|)+|Fcalc|, map: 2m|Fextr|-D|Fcalc|, ϕ_model).
.all_maps = False
calculate qFextr, Fextr, qFgenick, Fgenick, q_Fextr_calc, Fextr_calc
.only_qweight = False
In combination with Fextrapolated_map_types or all_maps, calculate all extrapolated structure factors and maps with q-weighting
.no_qweight = False
In combination with Fextrapolated_map_types or all_maps, calculate all extrapolated structure factors and maps without q-weighting
.fast_and_furious = False
Run fast and furious (aka without supervision). Will only calculate qFextr and associated maps, use highest peaks forα/occupancy determination (α/occupancy will be nonsense if map_explorer parameters being bad), run refinement with finally with derived α/occupancy. Not recommended but can be usefull for a first quick evaluation.
.negative_and_missing = *truncate_and_fill truncate_no_fill fref_and_fill
fref_no_fill fcalc_and_fill fcalc_no_fill fill_missing no_fill reject_and_fill reject_no_fill massage_and_fill massage_no_fill zero_and_fill zero_no_fill Handling of negative and missing extrapolated reflections (note that this will not be applied on FoFo difference maps). Please check the manual for more information. This parameters is NOT applicable for (q)Fgenick because negative reflections are rejected anyway. For refinement, default phenix.refine or refmac handling of negative/missing reflections is applied.

map_explorer
.threshold = 3.5
Integration threshold (in sigma)
.peak = 4.0
Peak detection threshold (sigma)
.radius = None
Maximum radius (A) to allocate a density blob to a protein atom in map explorer. Resolution will be used if not specified.
.z_score = 2.0
Z-score to determine residue list with only highest peaks
.use_occupancy_from_distance_analysis = False
Use occupancy from determination based on the differences between model_pdb and real-space refined model (only in slow_and_curious mode) instead of map explorer

refinement
.run_refinement = True
Run the automatic refinements. Setting this parameter to False can be useful when a manual intervention is required before running the refinements. The Refiner.py script can be used to run the refinements and subsequent analysis afterwards.
.use_refmac_instead_of_phenix = False
Use Refmac for reciprocal space refinement and COOT for real-space refinement instead of phenix.refine and phenix.real_space_refine
.phenix_keywords
.target_weights
.wxc_scale = 0.5
see phenix.refine refinement.target_weights.wxc_scale
.wxu_scale = 1.0
phenix.refine refinement.target_weights.wxu_scale
.weight_selection_criteria

```
      .bonds_rmsd = None
       phenix.refine refinement.target_weights.weight_selection_criteria.bonds_rmsd
      .angles_rmsd = None
       phenix.refine refinement.target_weights.weight_selection_criteria.angles_rmsd
      .r_free_minus_r_work = None
       phenix.refine refinement.target_weights.weight_selection_criteria.r_free_minus_r_work
  .refine
    .strategy = *individual_sites individual_sites_real_space rigid_body
     *individual_adp group_adp tls occupancies group_anomalous
     see phenix.refine refinement.refine.strategy
  .main
    .cycles = 5
     Number of refinement macro cycles for reciprocal space refinement
    .ordered_solvent = False
     Add and remove ordered solvent during reciprocal space refinement
  .map_sharpening
    .map_sharpening = False
       phenix map sharpening
  .real_space_refine
    .cycles = 5
       Number of refinement cycles for real space refinement
  .density_modification
    .density_modification = False
       use phenix.density_modification for density modification
.refmac_keywords
  .target_weights
    .weight = *AUTO MATRIx
     refmac WEIGHT
    .weighting_term = 0.2
     refmac weighting term in case of weight matrix
    .experimental_sigmas = *NOEX EXPE
     refmac use experimental sigmas to weight Xray terms
  .restraints
    .jelly_body_refinement = False
     run refmac ridge regression, also known as jelly body jelly body refinement. Slow refinement convergence, so take at least 50 refinement cycles.
    .jelly_body_sigma = 0.03
     sigma parameter in case of jelly body refinement ('RIDG DIST SIGM' parameter)
    .jelly_body_additional_restraints = None
     additional jelly body parameters (will be added to keyword RIDG )
    .external_restraints = None
     refmac external restraints (will be added to keyword external, e.g. harmonic residues from 225 A to 250 A atom CA sigma 0.02 )
  .refine
    .type = *RESTrained UNREstrained RIGId
     refmac refinement type refinement
    .TLS = False
     tls refinement before coordinate and B-factor refinement
    .TLS_cycles = 20
     number of TLS cycles in case of TLS refinement
    .bfac_set = 30
     reset individual B-factors to constant value before running TLS. Will only be applied in case TLS is run
    .twinning = False
     do refmac twin refinement
    .Brefinement = OVERall *ISOTropic
     refmac B-factor refinement
    .cycles = 20
     Number of refinement cycles for reciprocal space refinement
  .map_sharpening
    .map_sharpening = False
     refmac map sharpening
  .density_modification
    .density_modification = False
     use dm for density modification
    .combine = *PERT OMIT
     dm combine mode
    .cycles = 10
     number of dm cycles (ncycle keyword). Use only few cycles in case of combine =OMIT

output
 .outdir = None
  Output directory. Current directory directory will be used if not specified.
 .outname = None
  Output prefix (max 60 characters). Prefix of triggered_mtz will be used if not specified.
 .generate_φl_only = False
  Generate input φl-file and quit.
 .generate_fofo_only = False
  Stop Xtrapol8 after generation of Fourier Difference map
 .open_coot = True
  Automatically open COOT at the end.
 .ddm_scale = 1.5
  The ddm colors will range from -scale to +scale.
```

8 Reference

References

[1] Nicolas Coquelle, Michel Sliwa, Joyce Woodhouse, Giorgio Schirò, Virgile Adam, Andrew Aquila, Thomas R M Barends, Sébastien Boutet, Martin Byrdin, Sergio Carbajo, Eugenio De la Mora, R Bruce Doak, Mikolaj Feliks, Franck Fieschi, Lutz Foucar, Virginia Guillon, Mario Hilpert, Mark S Hunter, Stefan Jakobs, Jason E Koglin, Gabriela Kovacsova, Thomas J Lane, Bernard Lévy, Mengning Liang, Karol Nass, Jacqueline Ridard, Joseph S Robinson, Christopher M Roome, Cyril Ruckebusch, Matthew Seaberg, Michel Thepaut, Marco Cammarata, Isabelle Demachy, Martin Field, Robert L Shoeman, Dominique Bourgeois, Jacques-Philippe Colletier, Ilme Schlichting, and Martin Weik. Chromophore twisting in the excited state of a photoswitchable fluorescent protein captured by time-resolved serial femtosecond crystallography. *Nat Chem*, 10(1):31–37, 01 2018.

[2] Kevin Cowtan. dm. *Joint CCP4 and ESF-EACBM Newsletter on Protein Crystallography*, 31:34–38, 1994.

[3] Elke De Zitter, Nicolas Coquelle, Thomas R M Barends, and Jacques-Philippe Colletier. Xtrapol8. *Manuscript in preparation*, 2021.

[4] Ulrich K Genick. Structure-factor extrapolation using the scalar approximation: theory, applications and limitations. *Acta Crystallogr D Biol Crystallogr*, 63(Pt 10):1029–41, Oct 2007.

[5] Ralf W Grosse-Kunstleve, Nicholas K Sauter, Nigel W Moriarty, and Paul D Adams. The computational crystallography toolbox: crystallographic algorithms in a reusable software framework. *Journal of Applied Crystallography*, 35(1):126–136, 2002.

[6] Dorothee Liebschner, Pavel V Afonine, Matthew L Baker, Gábor Bunkóczi, Vincent B Chen, Tristan I Croll, Bradley Hintze, Li Wei Hung, Swati Jain, Airlie J McCoy, Nigel W Moriarty, Robert D Oeffner, Billy K Poon, Michael G Prisant, Randy J Read, Jane S Richardson, David C Richardson, Massimo D Sammito, Oleg V Sobolev, Duncan H Stockwell, Thomas C Terwilliger, Alexandre G Urzhumtsev, Lizbeth L Videau, Christopher J Williams, and Paul D Adams. Macromolecular structure determination using x-rays, neutrons and electrons: recent developments in phenix. *Acta Crystallogr D Struct Biol*, 75(Pt 10):861–877, Oct 2019.

[7] Z Ren, B Perman, V Srajer, T Y Teng, C Pradervand, D Bourgeois, F Schotte, T Ursby, R Kort, M Wulff, and K Moffat. A molecular movie at 1.8 a resolution displays the photocycle of photoactive yellow protein, a eubacterial blue-light receptor, from nanoseconds to seconds. *Biochemistry*, 40(46):13788–801, Nov 2001.

[8] T. Ursby and D. Bourgeois. Improved estimation of structure-factor difference amplitudes from poorly accurate data. *Acta Crystallogr. Sect. A*, 53:564–575, September 1997.

[9] Martyn D Winn, Charles C Ballard, Kevin D Cowtan, Eleanor J Dodson, Paul Emsley, Phil R Evans, Ronan M Keegan, Eugene B Krissinel, Andrew G W Leslie, Airlie McCoy, Stuart J McNicholas, Garib N Murshudov, Navraj S Pannu, Elizabeth A Potterton, Harold R Powell, Randy J Read, Alexei Vagin, and Keith S Wilson. Overview of the ccp4 suite and current developments. *Acta Crystallogr D Biol Crystallogr*, 67(Pt 4):235–42, Apr 2011.