Xtrapol8 Command Line manual

March 7, 2024

This is a manual to use Xtrapol8 v1.2.0 via command line.

Contents

1 Introduction

Xtrapol8 is software for the calculation of Bayesian-weighted Fourier difference (FoFo) maps, extrapolated structure factor amplitudes (ESFAs) and estimation of the occupancy. It is based on the cctbx toolbox [5] which come automatically with Phenix [6] and uses some CCP4 programs. [10] In order to run Xtrapol8, you will need to have a proper license 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.

This manual specifically concerns the usage of Xtrapol8 using the command line.

Please cite us if Xtrapol8 has been useful for your project: De Zitter, E., Coquelle, N., Oeser, P., Barends, T. R. M., Colletier, J.-P., Xtrapol8 enables automatic elucidation of low- occupancy intermediate-states in crystallographic studies, Communications Biology, *accepted*, https://doi.org/10.1038/s42003-022-03575-7.

2 Before you start

Clone or download the Xtrapol8 repository (`https://github.com/ElkeDeZitter/Xtrapol8.git`) 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 within 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-1.20-4459/phenix_env.sh
3. Add CCP4 to your PATH:
   In the same terminal source ccp4.setup-sh from the ccp4/setup-scripts or ccp4/bin folder (again you can use Finder to find the file).
   source /Applications/ccp4-8.0/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. e.g. on Mac:

5. Find out the shell you are using:
   echo $SHELL
   If your shell is */bin/bash*, then you should add the lines form step 2 and 3 to a file called ∼/.profile. If your shell is */bin/zsh*, then you should add the lines form step 2 and 3 to a file called ∼/.zprofile.

If you don't do this additional step, then take care to do step 1-4 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.
Processing consists of 5 steps (Figure 1):

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]

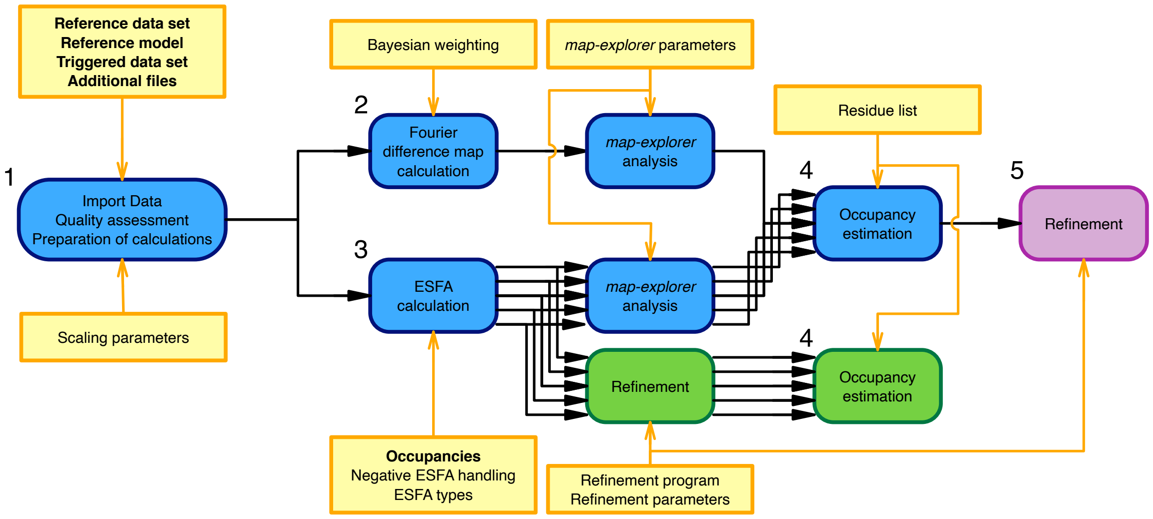

Figure 1: Principal steps taken by Xtrapol8. Main steps are depicted in blue, steps specific to the *slow and curious* mode in green, steps specific to the *fast and furious* mode in purple, input in yellow boxes with obligatory input indicated in bold.

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:
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 parameters 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 as default values will be used for undefined parameters. 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 9).

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

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

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.peak_integration_floor=3.5 map_explorer.peak_detection_threshold=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
```

Running Xtrapol8 with the slurm cluster and job management system

To launch Xtrapol8 with slurm, use the ”–wrap” argument to take care of properly importing the matplotlib Agg backend:

```
sbatch <sbatch-arguments> –wrap=”export MPLBACKEND=agg; phenix.python Fextr.py Xtrapol8.phil”
```

5 Running Xtrapol8 via the graphical user interface: XtrapolG8

Even though this manual concerns the usage of Xtrapol8 via the command line, in this section we give a quick introduction to the usage of the graphical user interface (GUI), XtrapolG8. but we refer to the dedicated manual for more information.

The GUI can be launched using the X8_GUI.py script, which you find in the folder with downloaded scripts.

```
phenix.python <where>/<to>/<find>/X8_gui.py
```

Just as for Xtrapol8, you can define an alias and store it in your ∼/.profile, ∼/.zprofile or ∼/.bashrc file:

```
alias XG8='phenix.python <where>/<to>/<find>/X8_gui.py
```

If you already have an Xtrapol8 input file, you might want to pass it through directly instead of loading all input files individually and manually setting all parameters. This can be done in two ways:

1. launch XtrapolG8 and open the input file from within the terminal:
   ```
   phenix.python X8_gui.py
   ```
   File → Open Phil

2. Add your input file to the command to launch XtrapolG8:
   ```
   phenix.python X8_gui.py Xtrapol8.phil
   ```

XtrapolG8 can also be used to inspect the output from a former run (being it from Xtrapol8 or XtrapolG8):

```
phenix.python X8_gui.py
```

File → Load Results

6 Parameters and Settings

6.1 Input

- **reference_mtz**
  Data for the ground/untriggered/unperturbed 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 process, 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 data usually contains the triggered state with a 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/unperturbed state in pdb or mmcif format. Preferably, this model has been obtained upon refinement with reference_mtz.

- **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 practice, 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 practice, this is determined by the common reflections between reference_mtz and triggered_mtz. We do not recommend the use of a low resolution cutoff unless you really know what you are doing.

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. It should be noted that the Bijvoet pairs will be averaged and merged during processing by Xtrapol8 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 the program truncate in CCP4 using French-Wilson scaling.

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.

It is thus very important to inspect your input files before using them in Xtrapol8 (e.g. using mtzdmp (CCP4), phenix.mtz.dump (phenix) or ViewHKL (CCP4)), and if necessary generate new input files that only contain your preferred columns (e.g. using cad (CCP4), phenix.reflection_file_converter (phenix) or reflection_file_editor (phenix GUI)).

6.2 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

6.3 Scaling

The b_scaling option is used to specify the scaling method of the triggered_mtz versus the reference_mtz dataset which will be carried out by scaleit:

- **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.

A specific resolution range for scaling the two data sets can optionally be given:

- **high_resolution**
- **low_resolution**

These resolution values will only be applied for scaling and will not be used to the truncate the data.

6.4 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 $|F_{obs}^{trig}| - |F_{obs}^{ref}|$ difference structure factors. More information can be found in Ursby and Bourgeois, 1997. [9] Q-weighting can be applied for the calculation of $|F_{obs}^{trig}| - |F_{obs}^{ref}|$ difference structure factors as well as for extrapolated structure factors.
- **k-weighting**
K-weighting is related Bayesian weighting scheme to reduce the contribution of uncertain structure factor values to the $|F_{obs}^{trig}| - |F_{obs}^{ref}|$ difference structure factors introduced by Ren *et al.* [7] K-weighting can be applied for the calculation of $|F_{obs}^{trig}| - |F_{obs}^{ref}|$ difference structure factors as well as for extrapolated structure factors. In the k-weighting scheme, the user has the option the specify the k-weight factor to be more or less stringent about outlier rejection (**kweight_scale**, default is 0.05).

6.5 FoFo types

Using the possibility of q-weighing, k-weighing and no weighing, three possible $|F_{obs}^{trig}| - |F_{obs}^{ref}|$ difference structure factors can be calculated which are used to calculate the Fourier difference map (Table 1):

- **qfofo**
Q-weighted $|F_{obs}^{trig}| - |F_{obs}^{ref}|$ difference structure factors. This type of weighted Fourier difference map is calculated by default.
- **fofo**
Non-weighted $|F_{obs}^{trig}| - |F_{obs}^{ref}|$ difference structure factors.
- **kfofo**
K-weighted $|F_{obs}^{trig}| - |F_{obs}^{ref}|$ difference structure factors.

Table 1: Three types of $|\mathsf{F}^{\mathsf{trig}}_{\mathsf{obs}}| - |\mathsf{F}^{\mathsf{ref}}_{\mathsf{obs}}|$ difference structure factors

type	Difference structure factors	Map Coefficients of Fourier difference map
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.
Using the keyword **output.generate_fofo_only**, Xtrapol8 can be set to stop running after the calculation of the Fourier difference map: *phenix.python Fextr.py <input file> output.generate_fofo_only=True*.

6.6 Extrapolated structure factors and map coefficients

Three different methods for calculation of extrapolated structure factors can be used. 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 Terwilliger and Berendzen, 1995. [8]

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**, **only_kweight** and **only_no_weight** allow an easy filtering 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**	
		$\mathbf{2F_{extrapolated} - F_{calc}}$ **type**	$\mathbf{F_{extrapolated} - F_{calc}}$ **type**
qfextr	$\mathrm{qF_{extr}} = \alpha \times \frac{q}{<q>} \times (\mathrm{F^{trig}_{obs}} - \mathrm{F^{ref}_{obs}}) + \mathrm{F^{ref}_{obs}}$	$2\mathrm{m}\lvert\mathrm{qF_{extr}}\rvert - \mathrm{D}\lvert\mathrm{F_c}\rvert, \phi_{\mathrm{ref}}$	$\mathrm{m}\lvert\mathrm{qF_{extr}}\rvert - \mathrm{D}\lvert\mathrm{F_c}\rvert, \phi_{\mathrm{ref}}$
kfextr	$\mathrm{kF_{extr}} = \alpha \times \frac{k}{<k>} \times (\mathrm{F^{trig}_{obs}} - \mathrm{F^{ref}_{obs}}) + \mathrm{F^{ref}_{obs}}$	$2\mathrm{m}\lvert\mathrm{kF_{extr}}\rvert - \mathrm{D}\lvert\mathrm{F_c}\rvert, \phi_{\mathrm{ref}}$	$\mathrm{m}\lvert\mathrm{kF_{extr}}\rvert - \mathrm{D}\lvert\mathrm{F_c}\rvert, \phi_{\mathrm{ref}}$
fextr	$\mathrm{F_{extr}} = \alpha \times (\mathrm{F^{trig}_{obs}} - \mathrm{F^{ref}_{obs}}) + \mathrm{F^{ref}_{obs}}$	$2\mathrm{m}\lvert\mathrm{F_{extr}}\rvert - \mathrm{D}\lvert\mathrm{F_c}\rvert, \phi_{\mathrm{ref}}$	$\mathrm{m}\lvert\mathrm{F_{extr}}\rvert - \mathrm{D}\lvert\mathrm{F_c}\rvert, \phi_{\mathrm{ref}}$
qfgenick	$\mathrm{qF_{genick}} = \alpha \times \frac{q}{<q>} \times (\mathrm{F^{trig}_{obs}} - \mathrm{F^{ref}_{obs}}) + \mathrm{F^{ref}_{obs}}$	$\mathrm{m_{ref}}\lvert\mathrm{qF_{genick}}\rvert, \phi_{\mathrm{ref}}$	$\mathrm{m_{ref}}\lvert\mathrm{qF_{genick}}\rvert - \mathrm{D}\lvert\mathrm{F_c}\rvert, \phi_{\mathrm{ref}}$
kfgenick	$\mathrm{kF_{genick}} = \alpha \times \frac{k}{<k>} \times (\mathrm{F^{trig}_{obs}} - \mathrm{F^{ref}_{obs}}) + \mathrm{F^{ref}_{obs}}$	$\mathrm{m_{ref}}\lvert\mathrm{kF_{genick}}\rvert, \phi_{\mathrm{ref}}$	$\mathrm{m_{ref}}\lvert\mathrm{kF_{genick}}\rvert - \mathrm{D}\lvert\mathrm{F_c}\rvert, \phi_{\mathrm{ref}}$
fgenick	$\mathrm{F_{genick}} = \alpha \times (\mathrm{F^{trig}_{obs}} - \mathrm{F^{ref}_{obs}}) + \mathrm{F^{ref}_{obs}}$	$\mathrm{m_{ref}}\lvert\mathrm{F_{genick}}\rvert, \phi_{\mathrm{ref}}$	$\mathrm{m_{ref}}\lvert\mathrm{F_{genick}}\rvert - \mathrm{D}\lvert\mathrm{F_c}\rvert, \phi_{\mathrm{ref}}$
qfextr_calc	$\mathrm{qF_{extr\_calc}} = \alpha \times \frac{q}{<q>} \times (\mathrm{F^{trig}_{obs}} - \mathrm{F^{ref}_{obs}}) + \mathrm{F^{calc}_{ref}}$	$2\mathrm{m}\lvert\mathrm{qF_{extr\_calc}}\rvert - \mathrm{D}\lvert\mathrm{F_c}\rvert, \phi_{\mathrm{ref}}$	$\mathrm{m}\lvert\mathrm{qF_{extr\_calc}}\rvert - \mathrm{DF_c}\rvert, \phi_{\mathrm{ref}}$
kfextr_calc	$\mathrm{kF_{extr\_calc}} = \alpha \times \frac{k}{<k>} \times (\mathrm{F^{trig}_{obs}} - \mathrm{F^{ref}_{obs}}) + \mathrm{F^{ref}_{calc}}$	$2\mathrm{m}\lvert\mathrm{kF_{extr\_calc}}\rvert - \mathrm{D}\lvert\mathrm{F_c}\rvert, \phi_{\mathrm{ref}}$	$\mathrm{m}\lvert\mathrm{kF_{extr\_calc}}\rvert - \mathrm{DF_c}\rvert, \phi_{\mathrm{ref}}$
fextr_calc	$\mathrm{F_{extr\_calc}} = \alpha \times (\mathrm{F^{trig}_{obs}} - \mathrm{F^{ref}_{obs}}) + \mathrm{F^{ref}_{calc}}$	$\mathrm{m}\lvert\mathrm{F_{extr\_calc}}\rvert - \mathrm{D}\lvert\mathrm{F_c}\rvert, \phi_{\mathrm{ref}}$	$\mathrm{m}\lvert\mathrm{F_{extr\_calc}}\rvert - \mathrm{D}\lvert\mathrm{F_c}\rvert, \phi_{\mathrm{ref}}$

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

The calculation of ESFAs can give a negative value. While they can be used to calculate electron density maps, these are not taken into consideration by structure refinement programs and thus lead to a lower apparent completeness. Therefore, several options are provided to treat these negatives:

- **Negative** The extrapolated structure factor amplitude distribution can be estimated using the truncate method, in which the extrapolated intensities are calculated by squaring the extrapolated structure factors and multiplication by the initial sign. Thereafter the ccp4 program truncate is run to recalculate extrapolated structure factor amplitudes. Otherwise the negative extrapolated structure factor amplitudes can be replaced by their observed or calculated reference counterparts ($\mathrm{F^{ref}_{obs}}$ or $\mathrm{F^{calc}_{obs}}$), rejected, or set to zero. There is also an option to don't treat them ("keep") in which case they are written to the output files as such. Since Genick suggests to reject the negative ESFAs [4], they will be rejected for "fgenick" type of ESFAs (independent of the method selected and other ESFA types).
- **Missing** In addition, missing reflections can be estimated (filled) upon map calculations. We refer to the Xtrapol8 publication for more information and suggested strategies to deal with the negative and missing reflections. [3]. The options "and_fill" and "no_fill" allows the user to specify whether or not missing reflections should be estimated to calculate electron density maps.

Electron density maps with the "keep_no_fill" are calculated in any case and can be found in the "maps-keep_no_fill" subfolders (see section 7.2)

6.7 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 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 main 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 Xtrapol8 in *fast and furious* mode, use *phenix.python Fextr.py <input file> f_and_maps.fast_and_furious=True*

6.8 Occupancy estimation

Xtrapol8 has several build-in methods to estimate the occupancy of the triggered state. All methods that are compatible with the input parameters will be run and results shown, but only one will be used for further steps within an Xtrapol8 run:

1. *Difference map* method: **difference map maximization**
   The *difference map* method uses the Fourier difference ($|F^{trig}_{obs}| - |F^{ref}_{obs}|$) and extrapolated difference maps ($|F_{extrapolated}| - |F_{calc}|$, Table 2 last column) to estimate the occupancy of the triggered state. Practically, this is done by *map-explorer*, which needs the following parameters:

   - **peak_detection_threshold**
     Peak detection threshold, expressed in sigmas (r.m.s.d.). Only peaks with an absolute value equal or above this value will be integrated.
   - **peak_integration_floor**
     Floor value for blob integration, expressed in sigmas (r.m.s.d.) (blob = peak from with more than 2 voxels). Peaks will be integrated from their maximum value towards this lower bound to avoid integration of noise.
   - **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:

   i Using the peak_detection_threshold parameter, we select the peaks in the $|F^{trig}_{obs}| - |F^{ref}_{obs}|$ map that are within the radius of an atom in the reference model.

   ii Integrate the $|F^{trig}_{obs}| - |F^{ref}_{obs}|$ peaks from their maximum value down to the peak_integration_floor (if we integrate to zero we'll get one big blob).

   iii Make a histogram of the $|F^{trig}_{obs}| - |F^{ref}_{obs}|$ peaks (they should have a normal distribution) and select only the highest using the z_score parameter. All map voxels outside of these peaks are masked in the following step.

   iv Using the mask, integrate the same peaks in the $|F_{extrapolated}| - |F_{calc}|$, sum the positive peaks, sum the negative peaks and sum of the absolute values, divide by the positive/negative/all peaks of the $|F^{trig}_{obs}| - |F^{ref}_{obs}|$ map and plot their values (green:positive, red:negative and black: sum of both) and search for the maximum of the sum.

2. *Difference map* method: **difference map Pearson CC**
   The second submethod of the *Difference map* method is a standard Pearson correlation coefficient between the $|F^{trig}_{obs}| - |F^{ref}_{obs}|$ map and the $|F^{trig}_{obs}| - |F^{ref}_{obs}|$ maps. Here, the complete map is used over the complete asymmetric unit, no selection of peaks. This should generate a very similar result to the first plot unless artifacts, such as big peaks in the solvent channel, appear. For now it is not possible to select the occupancy that is given by this plot, except in case of failure to estimate the occupancy based on the maximization of the peaks fails.

3. **Distance analysis** method
   In case of running Xtrapol8 in *calm and curious* mode with running the refinement steps, another 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 7).

   Also in the *distance analysis* method, the residue list arriving from the largest Fourier difference map peaks (residlist_Zscore<z-score>.txt) as found by map-explorer will be used. Hence, even though the maps are not used to estimate the occupancy, the map-explorer parameters (**peak_detection_threshold**, **peak_integration_floor**, **radius** and **z_score**) should be properly set to select the most useful amino acid residues.

   The *difference distance* method can be useful in cases where the $|F^{trig}_{obs}| - |F^{ref}_{obs}|$ does not show high and pronounced peaks, but can be very dependent on the refinement parameters used. Therefore, an

additional script is provided **refiner.py** in which reciprocal and real space refinement parameters can be altered and refinement can be run again. Currently, refiner.py only works using phenix.refine and phenix.real_space_refine. Refiner.py takes an input file which requires the Xtrapol8_out.phil file (see section 7) and an input file for reciprocal and real space refinement. Just as for Xtrapol8, it can be launched using phenix.python. To get the available parameters:
phenix.python refiner.py

The occupancy determination using the *difference map* and *distance analysis* methods can be re-run after a complete Xtrapol8 run in a standalone mode in which map-explorer parameters can be altered or custom residue list can be provided (**differencemap_analysis.py**, **distance_analysis.py**). The scripts do not use an input file and all arguments have to be passed through by the command line. To get all options:
phenix.python differencemap_analysis.py
phenix.python distance_analysis.py

6.9 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:
phenix.python Fextr.py <input file> refinement.run_refinement=False
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 as explained above.

- **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 6.7). 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, and rerun the automatic refinements using **refiner.py** is necessary.

6.10 Output

- **outdir**
  Directory in which the output will be stored. If not specified, the output will appear in a new directory called "Xtrapol8". If the directory already exists, Xtrapol8 will create a new one using the specified outdir name followed by a number. When launching Xtrapol8 runs rapidly one after the other, the automatic recognition of already existing output directories will fail and multiple Xtrapol8 runs will try to write in the same directory. This might give errors or can pass unnoticed, giving ambiguous results. We thus advice users to to change the name of the output directory argument before launching a new run. This is also important if Xtrapol8 runs are launched via a job scheduling system (e.g. slurm).

- **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 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 Fourier difference map, extrapolated and refined density maps for the estimated occupancy, the input model and the refined models for the estimated occupancy. The COOT script can also be launched afterwards using
  coot - - script <path/to/Xtrapol8/output/directory/subdirectory>/coot_all_<extrapolated structure factor type>.py 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.

- **ddm_scale**
  Scale factor for the ddm plot (see See 7.1). The ddm colors will range from -scale to +scale.

- GUI
  This should never be changed by the user. It will be changed by XtrapolG8 and is required for its correct functioning.

7 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 refiner.py script will need this Xtrapol8_out.phi file as argument in order to correctly find all files.

- **<outname>_m<weighting>FoFo**
  The $|F^{trig}_{obs}| - |F^{ref}_{obs}|$ difference map coefficients in three formats: mtz, ccp4. Xplor-files are not longer calculated from Xtrapol8 version 1.2.0 onwards.
  See Table 3 for column names in the mtz-file.

- The output files from map-explorer:
  - **peakintegration.txt**
    Contains the summary of the $|F^{trig}_{obs}| - |F^{ref}_{obs}|$ difference map peaks analysis (from map-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 7.1

- 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 7.2

- **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

7.1 Figures in the output directory

Figures are generated in pdf and png format.

- **Riso_CCiso.png**
  Plot showing Riso and CCiso of reference_mtz and triggered_mtz versus resolution (Figure 2).
  An overall Riso value below 10 % indicates that the two data sets are strongly 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.

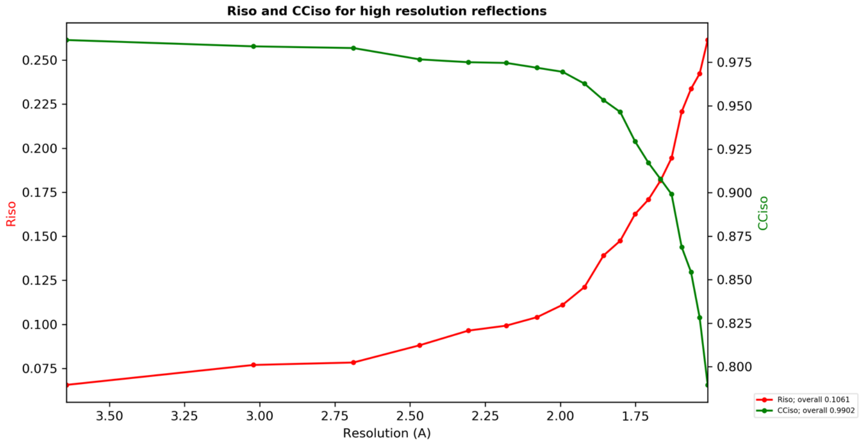

Figure 2: Example of plot showing the average Riso and CCiso values versus resolution. In this example, a high resolution cutoff at 1.9 Å is advised.

- **q_estimation.png**
  Plot showing the average q values and its range versus resolution. (Figure 3)

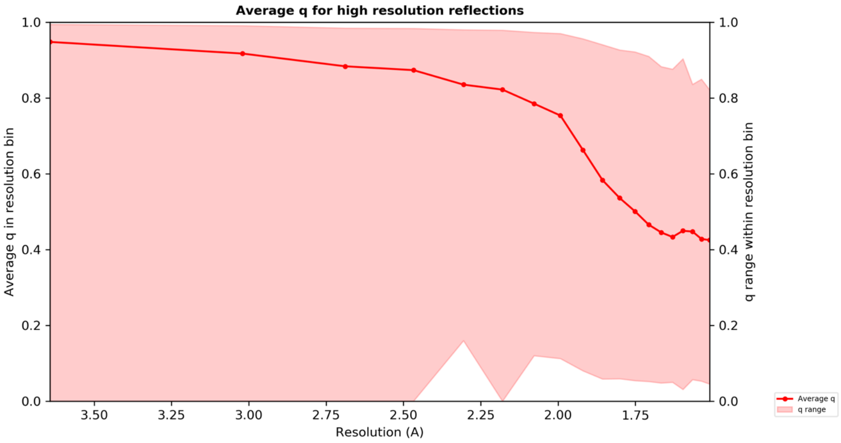

Figure 3: Example of plot showing average q value and its range versus resolution. In this example, the weight of the heigh resolution reflections is reduced as compared to the low resolution reflections.

- **k_estimation.png**
  Plot showing the average k-values and its range versus resolution. (Figure 4).

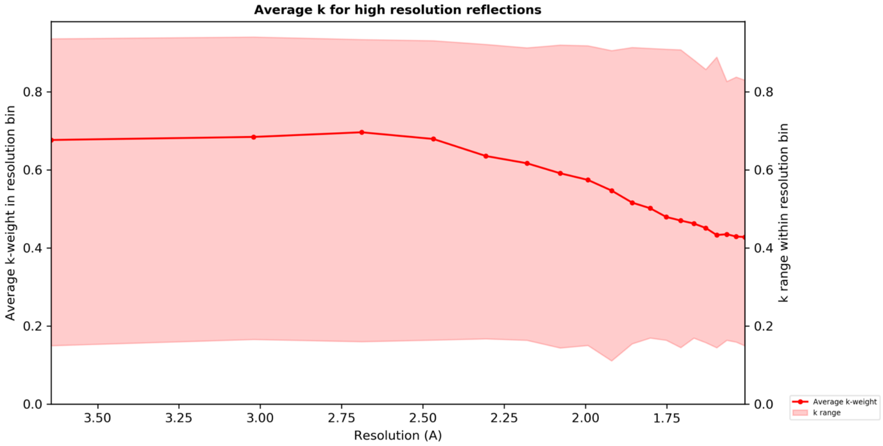

Figure 4: Example of plot showing average k value and its range versus resolution.

- **summed_difference_peaks.png**
  Plot showing the positive and negative integrated peak area versus amino acid and secondary structure (Figure 5). α-helices are depicted as pink bars whereas β sheets are depicted by blue triangles; positive and negative peaks in the $|F_{obs}^{trig}| - |F_{obs}^{ref}|$ 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.

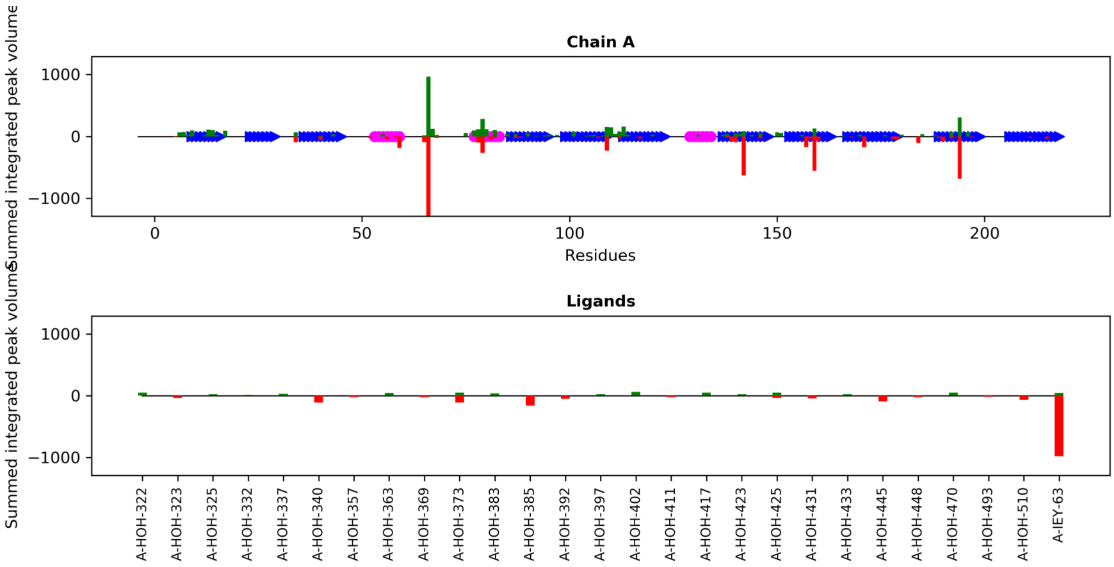

Figure 5: Example of secondary structure plot indicating the $|F^{trig}_{obs}| - |F^{ref}_{obs}|$ difference map peaks. The example model has four chains and several ligands and water molecules to which difference map peaks are annotated.

- **<q/k>FoFo_sigmas.png**
  Plot showing the the average $|F^{trig}_{obs}| - |F^{ref}_{obs}|$ 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 6).

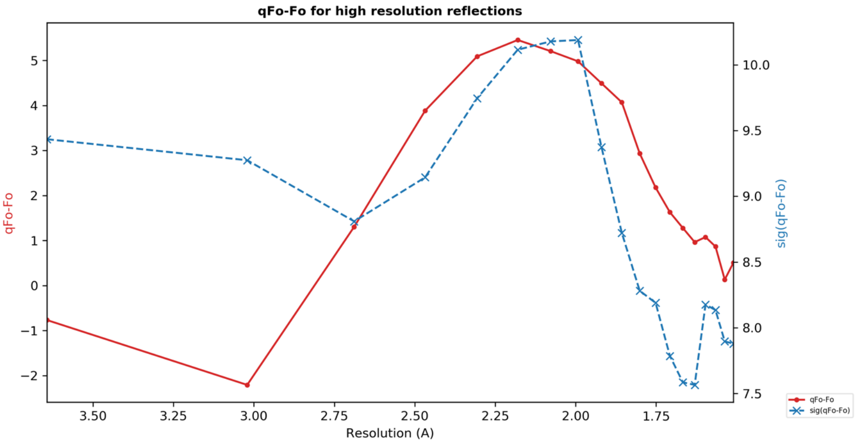

Figure 6: Example of plot showing the average $|F^{trig}_{obs}| - |F^{ref}_{obs}|$ structure factors and estimated errors. In this example, there are more negative differences than positive differences in each high resolution shell. The increase in sigma values at high resolution indicates that the sigma values are no longer correctly estimated

- **<extrapolated structure factor type>_sigmas.png**
  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 7).

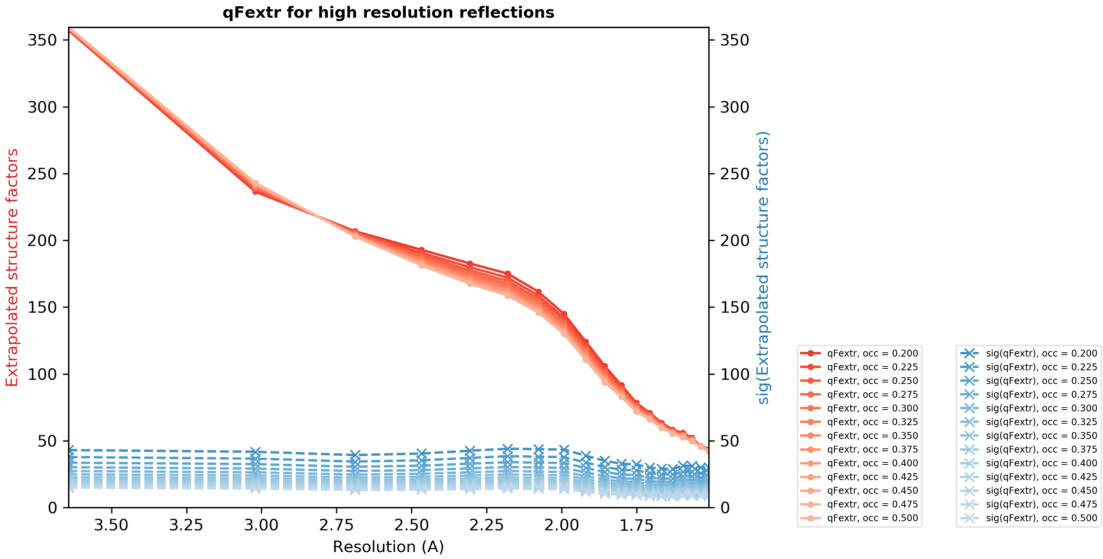

Figure 7: Pot showing the extrapolated structure factors and estimated errors for each type of extrapolated structure factors and occupancy. The bottom plot shows that the data strength decreases fast but at all resolution shells remain higher than the error values.

- **Neg_Pos_reflections_<extrapolated structure factor type>.png**
  Plot showing the absolute number (spheres, left axis) and percentage (right axis) of negative and positive extrapolated structure factors versus occupancy. This plot is generated for each type of requested extrapolated structure factors (Figure 8).

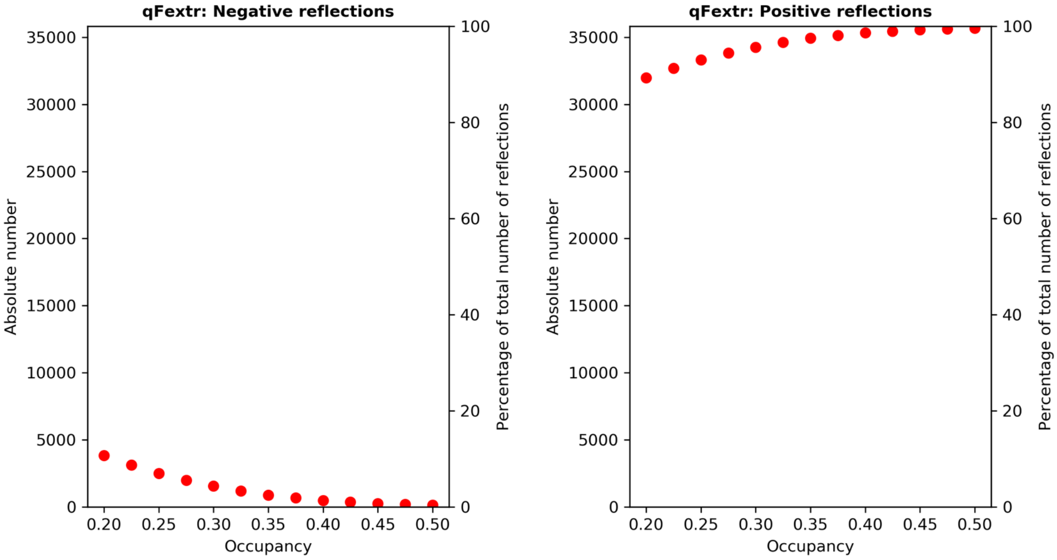

Figure 8: Example of plot showing the number of negative extrapolated structure factors versus the occupancy.

- **alpha_occupancy_determination_<extrapolated structure factor type>.png**
  Plot showing the normalized comparison of the extrapolated difference map peaks in order to estimate α and the occupancy, based the selected residues (residlist_Zscore<z-score>.txt), all residues and the selected ones with an enhanced signal-to-noise ratio (Figure 9). The α/occupancy with a normalized ratio of one, is the most likely α/occupancy, but manual inspection should be performed.
  A different picture is generated for each type of requested extrapolated structure factors.

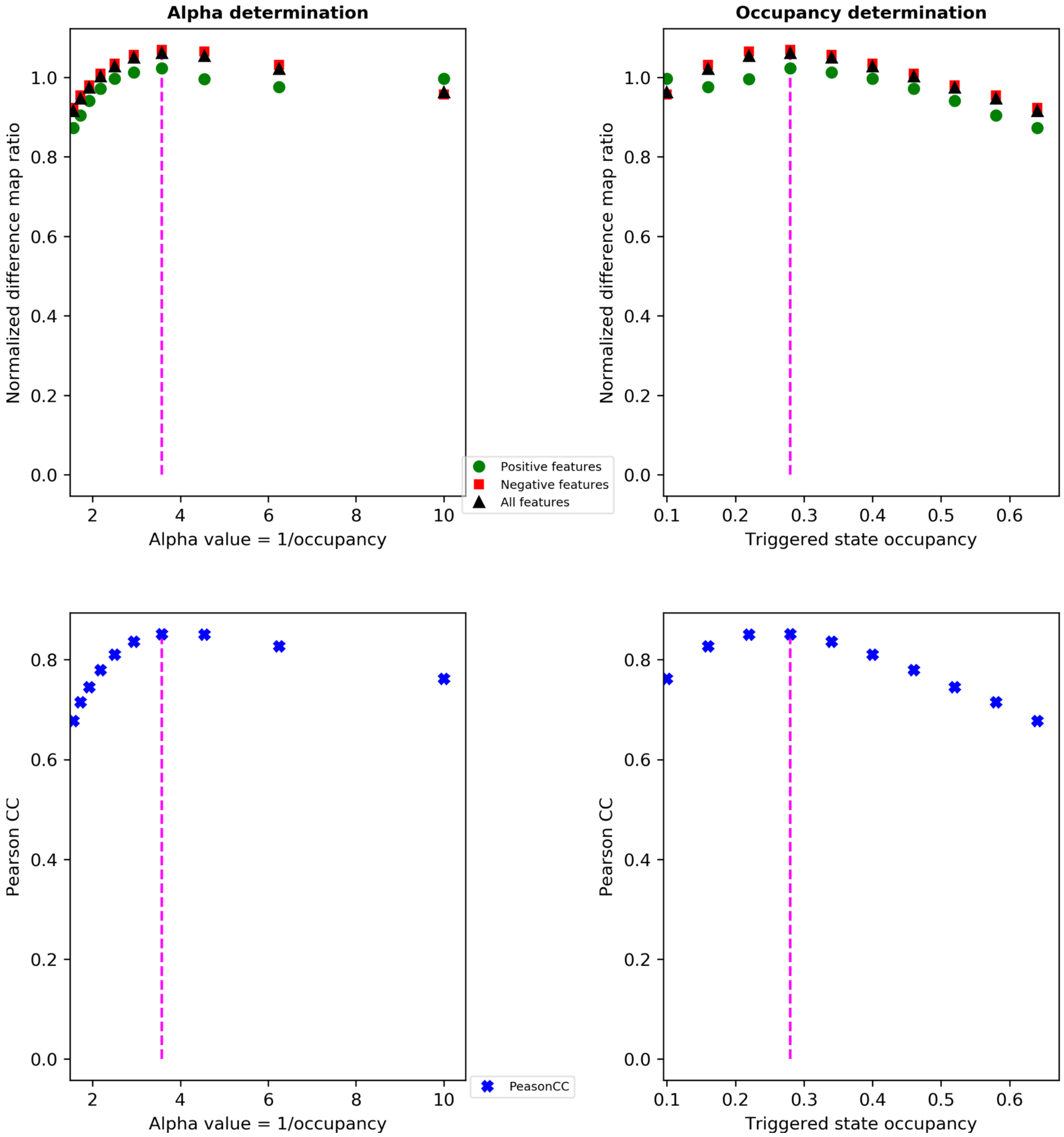

Figure 9: Example of plot showing the results of the *difference map* analysis. The top plots show the $|\mathrm{F}_{\mathrm{extrapolated}}| - |\mathrm{F}_{\mathrm{calc}}|$ over $|\mathrm{F}^{\mathrm{trig}}_{\mathrm{obs}}| - |\mathrm{F}^{\mathrm{ref}}_{\mathrm{obs}}|$ ratio of selected peaks versus occupancy and α value for a single extrapolated structure factor type. The bottom plot shows the Pearson correlation coefficient between the $|\mathrm{F}_{\mathrm{extrapolated}}| - |\mathrm{F}_{\mathrm{calc}}|$ and $|\mathrm{F}^{\mathrm{trig}}_{\mathrm{obs}}| - |\mathrm{F}^{\mathrm{ref}}_{\mathrm{obs}}|$ maps versus occupancy and α value for a single extrapolated structure factor type. The pink dashed line indicates the maximum in each of the plots.

- **Distance_difference_plot_<extrapolated structure factor type>.png**
  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 atoms of the refined and reference pdb files are calculated for the residues in the list and plotted versus α. If an sigmoidal fit can be made, they are maintained as significant. If less than 40 distances are significant, they are all plotted, otherwise they are not. The α is estimated based on the average of the fits of all distances at 99% of reaching the plateau level. Further, the average of all significant distances is plotted and an sigmoidal curve fitted and is used as an alternative α estimations.
  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.

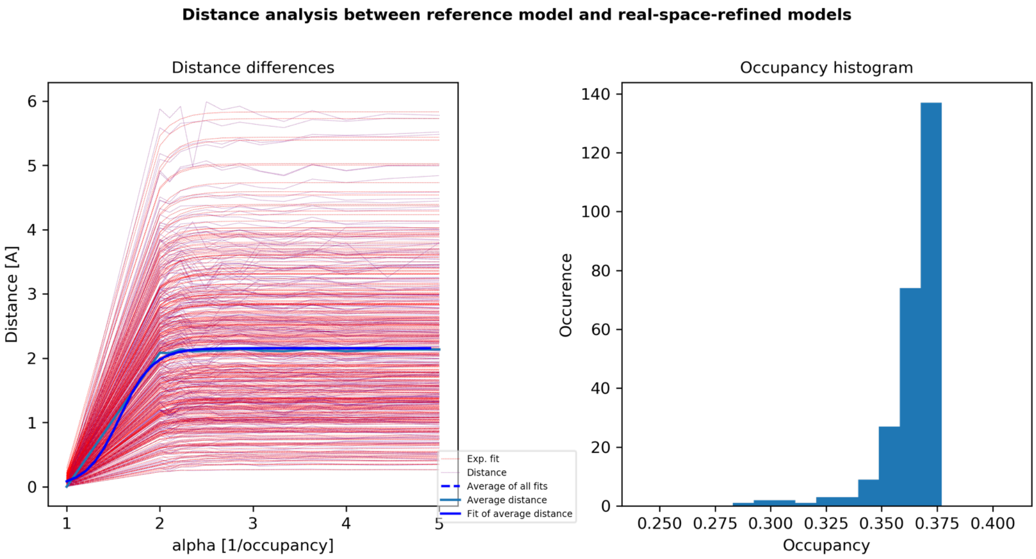

Figure 10: Example of plot showing the distances between atoms in the real-space refined models with those of model_in versus α value for a single extrapolated structure factor type (left). The purple lines indicate the experimental distances, their fits are shown by red lines. The average experimental distance is indicated in blue-green, the average fit and fit of the average distances are shown in blue with a dashed and full line, respectively. In this case, the two latter curves superpose. The right curve depicts the distribution of occupancy values as calculated by the individual distances fits.

- **<extrapolated structure factor type>_refinement_R-factors_per_alpha.png**
  Plot showing the Rwork, Rfree and Rwork-Rfree gap after reciprocal space refinement for each occupancy.
  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.

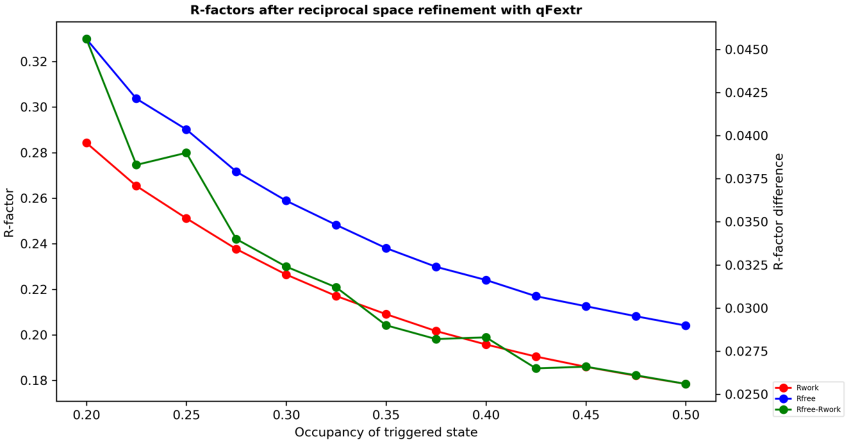

Figure 11: Example of plot showing the refinement R-factors after reciprocal space refinement versus occupancy for a single extrapolated structure factor type.

7.2 Subdirectory per occupancy and Bayesian weighting strategy

Each subdirectory from the output directory contains the output specific to a tested occupancy and whether or not q/k-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 Table 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**
    $2F_{\mathrm{extrapolated}} - F_{\mathrm{calc}}$ and $F_{\mathrm{extrapolated}} - F_{\mathrm{calc}}$ type map coefficients associated to a certain map type in mtz-format.
    See Table 2 for calculation of the map coefficients. See Table 3 for column names in the mtz-file.
  - **<outname>_occ<occupancy>_2<extrapolated structure factor type>-DFc.ccp4**
    $2F_{\mathrm{extrapolated}} - F_{\mathrm{calc}}$ 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**
    $F_{\mathrm{extrapolated}} - F_{\mathrm{calc}}$ type map coefficients associated to a certain map type in ccp4-format.
    See Table 2 for calculation of the map coefficients.

- **maps-keep_no_fill**
  This folder contains the $2mF_{\mathrm{extrapolated}} - DF_{\mathrm{calc}}$ and $mF_{\mathrm{extrapolated}} - DF_{\mathrm{calc}}$ electron density maps using the "keep_no_fill" strategy. This means that negatives are not treated and missing reflections are not filled. The figure of merit has been updated ("m" is used, and not "m_{ref}").

- Figures
  Three figures are generated for each type of requested extrapolated structure factors and occupancies. The first two figures give 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.png**
    Plot showing the absolute number (red, left axis) and percentage (blue, right axis) of negative extrapolated structure factors versus resolution, and plot showing the completeness versus resolution before data was manipulated to eliminating negative extrapolated structure factors (Figure 12). "True completeness" is the completeness of the positive extrapolated structure factors only. A completeness/True completeness above 90% is highly recommended.

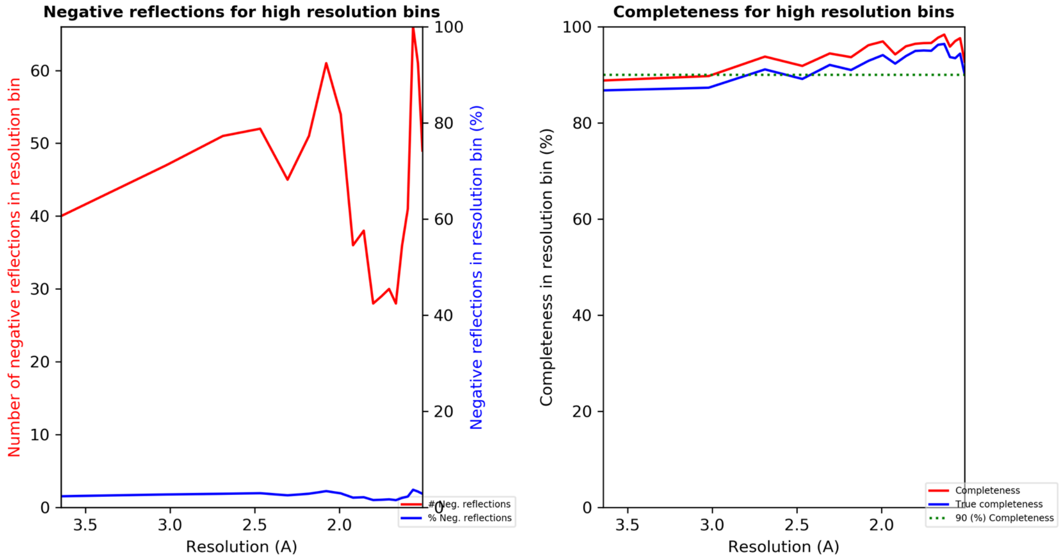

Figure 12: Example of plot showing negative extrapolated structure factors versus resolution on the right, and plot showing the completeness and true completeness on the right.

  - **<extrapolated structure factor type>_occupancy<occupancy>_FsigF.png**
    Plot showing the average extrapolated structure factors over the estimated error versus resolution (Figure 13). $\langle F/\sigma(F)\rangle$ above 1.22 is highly recommended ($\sim \langle I/\sigma(I)\rangle$ >1.5).

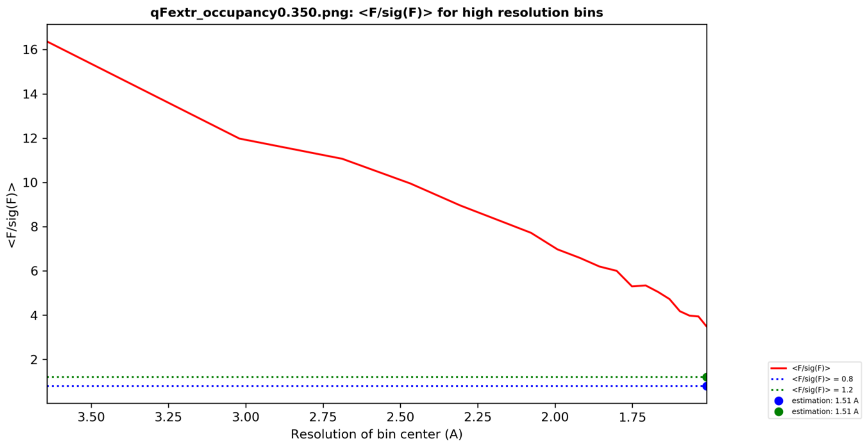

Figure 13: Example of plot showing $\langle F/\sigma(F)\rangle$ versus resolution. In this example, the resolution cutoff is badly estimated because the signal-to-noise ratio is high in each resolution shell and hence the full resolution range can be used.

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

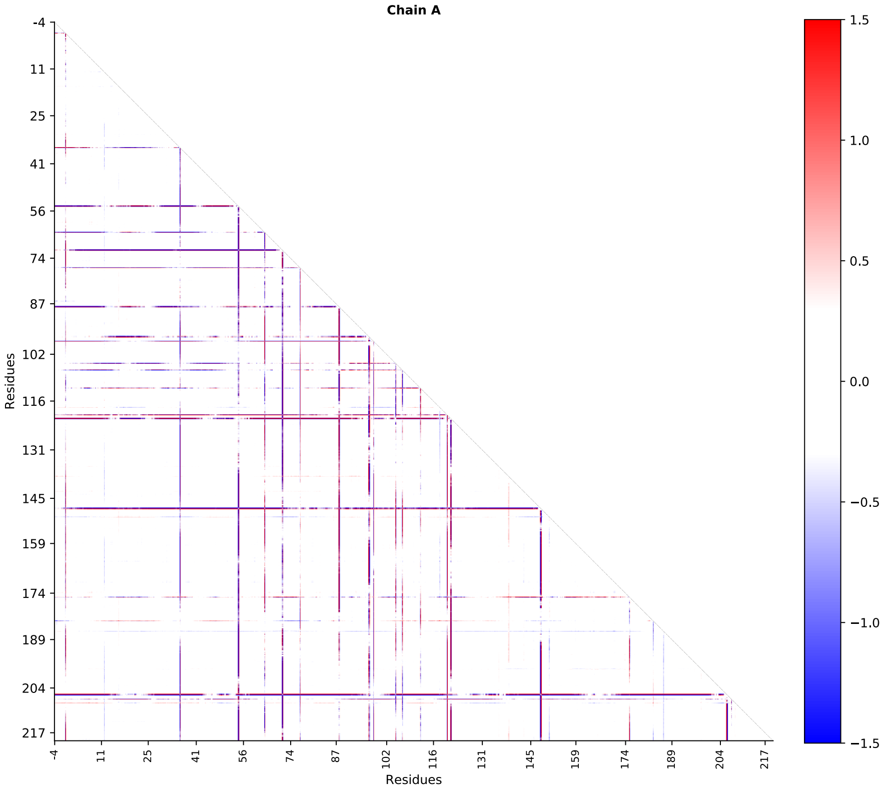

Figure 14: Example of plot showing a distance difference matrix (ddm) indicating for each protein chain the main atomic differences between the input model and the model (reciprocal space + real space refined) at the estimated occupancy.

- Output of refinements
  In case of running Xtrapol8 in *fast and furious* mode, the following files will only be present in the folder

Table 3: mtz column labels

type	**Difference structure factors**		**Map Coefficients**	
	amplitudes	**sigmas**	**amplitudes**	**phases**
qfofo	QFDIFF	SIGQFDIFF	QFOFOWT	PHIQFOFOWT
fofo	FDIFF	SIGFDIFF	FOFOWT	PHIFOFOWT
kfofo	KFDIFF	SIGKFDIFF	KFOFOWT	PHIKFOFOWT
type	**Extrapolated structure factors**		**Map Coefficients***	
	amplitudes	**sigmas**	**amplitudes**	**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

* The same logic is applied for creation of the columns for the $F_{\text{extrapolated}} - F_{\text{calc}}$ difference map.

of the estimated occupancy. In *slow and curious* mode this output 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 as input.
- **<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, using the extrapolated electron density map and reference_pdb as input.
- **<outname>_occ<occupancy>_<structure factor type>**
  -DFc_reciprocal_space_real_space.pdb or
  <outname>_occ<occupancy>_<structure factor type>
  -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 unsuccessful (in the latter case the pdb files from the simple reciprocal or real-space refinement are used).

- **coot_all_<extrapolated structure factor type>.py**
  Script to open all relevant models and maps in Coot for a certain type of extrapolated structure factors.
  coot - -script coot_all_<extrapolated structure factor type>.py
  This script is only present in 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 a successful Xtrapol8 run.

8 Trouble shooting

Below you can a list of known errors and bugs. The exact error message can slightly change depending on the version, local paths, given Xtrapol8 arguments, etc.

- *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.
  → Make sure you have the permission to create the output directory (make sure you have write permission).

- *File "/Applications/phenix-1.19-4080/modules/cctbx_project/mmtbx/bulk_solvent/f_model_all_scales.py", line 177, in update_solvent_and_scale_2. assert approx_equal(self.r_all(), r_all_from_scaler). AssertionError*
  This happens upon calculation of F_model, most probably during the calculation of extrapolated electron density maps. This might be caused by a low data set completeness as a result of rejection of many negative structure factors.
  → Check if there are lot of negative extrapolated structure factors and alter your strategy to handle the negative extrapolated structure factors or increase the lowest occupancy. Note that when the number of negatives is high for a certain set of extrapolated structure factors, truncate will fail and the strategy for that set will be changed to rejecting the negatives, leading to rejection of a large fraction of structure factors and hence failing of the subsequent scaling algorithm.

- *SyntaxError: Non-ASCII character '\xe2' in file /mntdirect/_programs/x86_64-linux/phenix/1.20.1-4487/phenix-1.20.1-4487/conda_base/lib/python2.7/site-packages/scipy/stats/_continuous_distns.py on line 3346, but no encoding declared; see http://python.org/dev/peps/pep-0263/ for details*
  This can happen on Linux systems and Phenix 1.20. It is caused by a unicode issue in scipy. As Xtrapol8 uses phenix.python and thus the python packages that are shipped with Phenix. There are three possible solutions. Not all might work for you and for the last two you might need administrator (sudo) rights:
  → Use Phenix version 1.19 instead of version 1.20.
  → Install scipy 1.2.1 within your phenix.python environment: *phenix.python -m pip install scipy==1.2.1*
  → Add the following line to the top of the listed scipy scripts: *# -*- coding: utf-8 -*-*

- *Traceback (most recent call last):*
  File "/Fextr.py", line 3401, in ¡module¿
  run(sys.argv[1:])
  File "/Fextr.py", line 2931, in run
  if len(os.listdir(new_dirpath_q)) == 0:
  OSError: [Errno 2] No such file or directory: 'qweight_occupancy_0.100'
  This happens if two (or more) Xtrapol8 runs are writing in the same output directory. This is a consequence of launching two (or more) Xtrapol8 shortly one after the other, causing the automatic check for the existence of output direct to fail.
  → This error can be avoided by changing the output directory from run to run, or wait longer between launching two Xtrapol8 runs.

9 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 using scaleit.
.high_resolution = None
High resolution for scaling the triggered data vs reference data using scaleit (Angstrom). Will only be used if high resolution of the input data files extends to this value. This only implies scaling, the data will not be cut. If not specified, then input.high_resolution will be used.
.low_resolution = None
Low resolution for scaling triggered data vs reference data using scaleit (Angstrom). Will only be used if low resolution of the input data files extends to this value. This only implies scaling, the data will not be cut. If not specified, then input.low_resolution will be used.

f_and_maps
.fofo_type = *qfofo fofo kfofo
Calculate q-weighted, non-weighted or k-weighted Fourier difference (Fo-Fo) map.
.kweight_scale = 0.05
scale factor for structure factor difference outlier rejection in k-weigting scheme.
.f_extrapolated_and_maps = *qfextr fextr kfextr qfgenick fgenick kfgenick qfextr_calc fextr_calc kfextr_calc
Extrapolated structure factors and map types:
qFextr, kFextr, Fextr: (q/k-weighted)-Fextr structure factors and maps by Coquelle method:
$((\mathrm{q/k})\mathrm{F}_{\mathrm{extr}} = \alpha \times (\frac{q/k}{<q/k>}) \times (\mathrm{F}^{\mathrm{trig}}_{\mathrm{obs}} - \mathrm{F}^{\mathrm{ref}}_{\mathrm{obs}}) + \mathrm{F}^{\mathrm{ref}}_{\mathrm{obs}}$, map: $2\mathrm{m}|(\mathrm{q/k})\mathrm{F}_{\mathrm{extr}}| - \mathrm{D}|\mathrm{F}_{\mathrm{c}}|, \phi_{\mathrm{ref}})$.
qFgenick, kFgenick, Fgenick: (q/k-weighted)-Fextr structure factors and maps by Genick method:
$((\mathrm{q/k})\mathrm{F}_{\mathrm{genick}} = \alpha \times (\frac{(q/k)}{<q/k>}) \times (\mathrm{F}^{\mathrm{trig}}_{\mathrm{obs}} - \mathrm{F}^{\mathrm{ref}}_{\mathrm{obs}}) + \mathrm{F}^{\mathrm{ref}}_{\mathrm{obs}}$, map: $\mathrm{m}_{\mathrm{dark}}|(\mathrm{q/k})\mathrm{F}_{\mathrm{genick}}|, \phi_{\mathrm{ref}}$
qFextr_calc, kFextr_calc, Fextr_calc: (q-weighted)-Fextr structure factors and maps by Fcalc method:
$((\mathrm{q/k})\mathrm{F}_{\mathrm{extr\_calc}} = \alpha \times (\frac{q/k}{<q/k>}) \times (\mathrm{F}^{\mathrm{trig}}_{\mathrm{obs}} - \mathrm{F}^{\mathrm{ref}}_{\mathrm{obs}}) + \mathrm{F}^{\mathrm{calc}}_{\mathrm{ref}}$, map: $2\mathrm{m}|(\mathrm{q/k})\mathrm{F}_{\mathrm{extr\_calc}}| - \mathrm{D}|\mathrm{F}_{\mathrm{c}}|, \phi_{\mathrm{ref}}$.
.all_maps = False
Calculate all extrapolated structure factors and maps.
.only_qweight = False
Calculate all extrapolated structure factors and maps with q-weighting.
.only_kweight = False
Calculate all extrapolated structure factors and maps with k-weighting.
.only_no_weight = False
Calculate all extrapolated structure factors and maps without q/k-weighting.
.fast_and_furious = False
Run fast and furious (aka without supervision). Will only calculate qFextr and associated maps and run refinement with finally with derived alpha/occupancy. Default parameters will be used for fofo_type and negative_and_missing. 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 keep_and_fill keep_no_fill reject_and_fill reject_no_fill zero_and_fill zero_no_fill
Handling of negative and missing extrapolated structure factor amplitudes (ESFAs) Note that this will not be applied on the Fourier difference map. If selected, filling of missing reflections is only carried on maps of the 2mFextr-DFcalc type. This parameters is NOT applicable for (q/k)Fgenick because negative reflections are rejected anyway. For refinement, default phenix.refine or refmac handling of negative/missing reflections is applied. keep_no_fill maps will be calculated in addition in all cases. keep_and_fill and keep_no_fill replace the old fill_missing and no_fill arguments which will become invalid keywords in future Xtrapol8 versions.

map_explorer
.peak_integration_floor = 3.5
Floor value for peak integration (sigma). Peaks will be integrated from their maximum value towards this lower bound to avoid integration of noise..
.peak_detection_threshold = 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.
.occupancy_estimation = *difference_map_maximization difference_map_PearsonCC distance_analysis
Select a main method for the occupancy estimation in Xtrapol8. Take care that the distance_analysis method can only be used in calm_and_curious

```
   mode.

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
       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
       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 (refinement.refine.main.ordered_solvent).
      .simulated_annealing = False
       Simulated annealing during refinement.
    .simulated_annealing
      .start_temperature = 5000
       start temperature for simulated annealing.
      .final_temperature = 300
       final temperature for simulated annealing.
      .cool_rate = 100
       cool rate for simulated annealing.
      .mode = every_macro_cycle *second_and_before_last once first first_half
       simulated annealing mode.
    .map_sharpening
      .map_sharpening = False
       phenix map sharpening.
    .additional_reciprocal_space_keywords = None
     Additional phenix.refine keywords which cannot be altered via included options (e.g. ncs_search.enabled=True).
     real_space_refine
      .cycles = 5
       Number of refinement cycles for real space refinement.
    .additional_real_space_keywords = None
     Additional phenix_real_space.refine keywords which cannot be altered via included options (e.g. ncs_constraints=False).
    .density_modification
      .density_modification = False
       use dm (ccp4) for density modification.
      .combine = PERT *OMIT
       dm combine mode.
      .cycles = 3
       number of dm cycles (ncycle keyword). Use a lot of cycles when combine=PERT and only few cycles when combine=OMIT
  .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.
    .additional_refmac_keywords = None
     Additional refmac keywords which cannot be altered via included options (e.g. ncsr local).
    .density_modification
      .density_modification = False
       use dm for density modification.
      .combine = PERT *OMIT
       dm combine mode.
      .cycles = 3
       number of dm cycles (ncycle keyword). Use a lot of cycles when combine=PERT and only few cycles when combine=OMIT

 output
  .outdir = None
   Output directory. Xtrapol8 be used if not specified.
  .outname = None
   Prefix or suffix for output files. The prefix of triggered_mtz
   will be used if not specified.
  .generate_phil_only = False
   Generate input phil-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.
  .GUI = False
   Xtrapol8 launched from GUI. In order to work correctly, this
   should never be manually changed.
```

10 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, Paula Oeser, Thomas R. M. Barends, and Jacques-Philippe Colletier. Xtrapol8 enables automatic elucidation of low-occupancy intermediate-states in crystallographic studies. *Communications Biology*, 5(1):640, 2022.

[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 C Terwilliger and J Berendzen. Difference refinement: obtaining differences between two related structures. *Acta Crystallogr D Biol Crystallogr*, 51(Pt 5):609–18, Sep 1995.

[9] 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.

[10] 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.