

EMAN

Tutorial Session #2

Steve Ludtke

CTF Parameter Determination
Refining your Model

Boxing

(aka Particle Selection)

- `cd ~/demo/samples/boxer`
- `boxer jj5339.f.mrc`

Tricks:

- `makeboxref.py groel.mrc sym=d7 invert ang=15`
- `batchboxer input=jj5346.f.mrc auto=.3,.7,.1 dbout=5346.box
refimg=best.hed`
- `alignhuge jj0881f.mrc jj0880f.mrc jj0881.ali.mrc`
- `focalpair jj0880f.mrc jj0881f.mrc 880.1.mrc im1=0880.init.hed
im2=0881.init.hed sffile=groel.sm filt=4`

Boxing

(aka Particle Selection)

- *cd ~/demo/groel/stage1*
- *batchboxer input=jj5337.f.mrc dbbox=5337.box
output=../stage2/5337.hed invert*

etc.

Initial Model (with symmetry)

- *startcsym (or startoct):*
- *cd ~/demo/groel/stage2*
- *proc2d 5337.hed orig.hed lp=14 ...*
- *cenalignint orig.hed maxshift=8 frac=0/4 ...*
- *startcsym ali.hed 50 sym=c7 [fixrot=90]*
- *volume threed.0a.mrc 2.8 set=800*
- *proc3d threed.0a.mrc masked.mrc automask2=15,.8,5*

Initial Model (with symmetry)

- *examine:*

avg.hed

classes.hed

sym.hed

threed.0a.mrc

masked.mrc

Initial Model (without symmetry)

- *startnrclasses and startAny:*
- *cd ~/demo/tough_cases/asymmetric/KIF/stage1*
- *startnrclasses start.hed 20*
- *examine classes.nr.hed, make good.hed*
- *startAny good.hed*

Basic Image Processing

- *iminfo*
- *proc2d*
- *proc3d*
- *procpdb.py*

CTF Parameter Determination

- *fitctf* - only if you have a real or simulated structure factor curve
- *ctfit* - check results, or fit if you don't have a structure factor

CTF Parameter Determination

```
cd ~/demo/groel/stage3
```

```
ctfit *hed
```

```
fitctf 5337.hed sf=groel.sm apix=2.8 V=400 Cs=4.1 acfix=.1
```

etc.

```
ctfit *.tnf
```

Model Refinement

cd ~/demo/groel/stage4

- *For the first iteration, the following was run:*

*refine 1 hard=25 ctfcw=groel.sm sym=d7 ang=4.28 mask=42
pad=128 classkeep=.5 classiter=8 xfiles=2.8,800,99 3dit=2
3dit2=2 amask=15,.7,5 shrink=2*

- *Look at:*

x?.mrc

classes.1.hed

proj.hed

cls.lst*

also: refine.log, particle.log

Refine Options

- Projection:

mask=<r>

proc=<n>

sym=<cn,dn,etc>

ang=<dang>

tree=<2,3>

perturb

- Classification:

sep=<n>

tree=<2,3>

shrink=<n>

maxshift=<rad>

precen, usefilt, slow

mask=<r>, imask=<r>

amask=<r>, <thr>, <iter>

proc=<n>

ctfc=<res>, ctfcw=<SF> or median

- Class Averages:

classiter=<n>

classkeep=<sig mult, but not 0>

mask=<r>, imask=<r>

amask=<r>, <thr>, <iter>

proc=<n>

ctfc=<res>, ctfcw=<SF> or median

maxshift=<r>

precen, slow

euler2=<oversmp>

Refine Options

- 3D Reconstruction:

hard=<phase err>

pad=<size>

sym=<cn,dn,etc>

3dit=<n>,3dit2=<n>

mask=<r>,imask=<r>

proc=<n>

collapse=<alt thr>

- Postprocessing (betw iter):

xfiles=<A/pix>,<mass in kd>,<ali to>

amask=<r>,<thr>,<iter>

filt3d=<lp rad>

Refine Options

REQUIRED:

- `<total iter>` - Final number of iterations you wish to be complete in this directory. ie - if 5 are complete and you specify 6, 1 more iteration will run
- `mask=<rad>` - A circular mask to be applied virtually everywhere. Should be a few pixels larger than the largest radius of your particle
- `ctfc=`, `ctfcw=` OR `median` - CTF correction options. `ctfc` takes a filter resolution in Angstroms, but `ctfcw` is far superior. `ctfcw` takes the name of a 1D structure factor file used when fitting the data (MUST be the same file). `median` does no CTF correction at all. Naturally the data must be properly preprocessed for this option to function, otherwise crashes or invalid results are likely.
- `hard=<phase err>` - Rather obscure. This option determines when a class-average should be excluded from the 3D reconstruction process. 25 is generally good. see 'make3d' for more info.
- `sym=<cn,dn,oct,icos>` - For asymmetric objects, either omit this specification, or use 'c1'. `cn` denotes a single n-fold rotational symmetry (about the z-axis), `dn` denotes n-fold dihedral symmetry (`cn` with n 2-folds in the x-y plane), `oct` is octahedral (2-3-4, symmetry of a cube), `icos` is icosahedral (2-3-5).
- `ang=<dang>` - Angular spacing between projections. Smaller numbers produce more projections, usually between 2-10.

Refine Options

REQUIRED:

- `pad=<size>` - This is used to reduce artifacts in Fourier reconstruction. Should be about 25% larger than your model in most cases, and close to a power of 2 or 3, ie - model size 50 -> `pad=64`, 70->96, 110->128
- `classkeep=<sig mult>` - This determines how many raw particles are discarded for each class-average. This is defined in terms of the standard-deviation of the self-similarity of the particle set. A value close to 0 (should not be exactly 0) will discard about 50% of the data. 1 is a typical value, and will typically discard ~10-20% of the data.
- `classiter=<n>` - Generation of class-averages is an iterative process. Rather than just aligning the raw particles to a reference, they are iteratively aligned to each other to produce a class-average representative of the data, not of the model. This is the number of iterations to use. 8 is typical. Less than 5 is rarely advisable. More than 8 is rarely useful.

Refine Options

OPTIONAL:

- `filt3d=<rad>` - Applies a lowpass filter to the 3D model between iterations. This can be used to correct problems that may result in high resolution terms being upweighted. `<rad>` is the same as for the `'lp='` option in `proc3d`.
- `sep=<n>` - This interesting option causes each particle to be assigned to the `n` best classes, not just the single best class. Generally this would be used with a small or negative `classkeep` value. This can help in cases where the data is very noisy and classification is ambiguous. It shifts the final classification decision to the class-averaging step.
- `tree=<2,3>` - This can be a risky option, but it can produce dramatic speedups in the refinement process. Rather than comparing each particle to every reference, this will decimate the reference population to 1/4 or 1/9 of its original size, classify, then locally determine which of the matches is best. It is safest in conjunction with very small angular steps, ie - large numbers of projections. The safest way to use this is either a) for high-resolution, small-ang refinement or b) for the initial iterations of refinement (then turn it off for the last couple of iterations).

Refine Options

TYPICAL (not required):

- `xfiles=<A/pix>,<mass>,<ali to>` - This is a convenience option. For each 3D model it will produce a corresponding x-file: `threed.1a.mrc -> x.1.mrc`. Based on `A/pix` and `mass` (in kd), the x-file will be scaled so an isosurface threshold of 1 will contain ~ the specified mass. 'ali to' is an iteration number. ie - if 'ali to' is 4, then `x.7.mrc` would be aligned in 3D to `x.4.mrc`. `x.3.mrc` would not be aligned at all. Often this is set to a large value, like 99.
- `3dit=<iter>, 3dit2=<iter>` - While not strictly required, these are virtually always used. They apply a real-space iterative reconstruction technique to the 3D model to clean up artifacts caused by Fourier techniques. Typically `3dit=1` and `3dit2=2`.
- `amask=<r>,<threshold>,<iter>` - This option applies an automatically generated 'form fitting' mask to the model after each iteration. The mask generation is generally quite good. See `proc3d` option `automask2` for details. This option can only be used in conjunction with `xfiles=`, since selection of the threshold requires proper volume normalization. This option can have a profound effect on proper convergence, but should be used with caution, and after convergence, this option should be disabled and 2 more iterations run to insure nothing important has been masked out.

Refine Options

TYPICAL (not required):

- `shrink=<n>` - Another option that can produce dramatic speed improvements. Strangely, in some cases, this option can actually produce an improvement in classification accuracy. This option scales the particles and references down by a factor of `n` before classification. Since data is often heavily oversampled, and classification is dominated by low resolution terms, this can be both safe, and actually improve classification by 'filtering' out high resolution noise. Models with Dn symmetry can experience iterative misclassification of the equatorial Euler angles. This option has been shown to correct that problem in several cases. Generally `shrink=2` is safe and effective. In cases of extreme oversampling, larger values may be ok.
- `euler2=<oversmp factor>` - This option should produce improvements in convergence and reconstruction quality, but has produced mixed results in the past. It adds an additional step to the refinement process in which class-averages orientations are redetermined by projection-matching. The parameter allows you to decrease the angular step (`ang=`) used to generate projections. ie - 2 would produce projections with angular step of `ang/2`. It may be worth trying, but use it with caution on new projects.

Refine Options

EXPERIMENTAL (not required):

- usefilt - This flag allows one to use arbitrarily filtered raw particles for classification purposes, but still use the unfiltered data when generating the actual reconstruction. To use this option, apply filter the data from start.hed into start.filt.hed.
- collapse=<alt max> - This option is used to correct for a problem wherein side views of particles with Dn symmetry can iteratively 'drift' from alt=90 up to smaller altitudes. The degree to which this can happen depends on noise levels in the particles. This experimental option will 'push' these values back to 90 degrees. It should always be followed with a couple of rounds of refinement without this option, to allow things to relax again.
- perturb - This is a very new option, which is potentially useful, and at worst should be harmless. However, it has not been well characterized yet. Rather than generating Euler angles at evenly spaced positions, it adds some randomness to the positions. This should produce a more uniform distribution of data in 3D Fourier space and reduce Fourier artifacts.
- imask=<rad> - This option was designed to improve classification for virus particles containing non-icosahedral DNA/RNA. It has not been tested in a long time and may or may not function.

Compile EMAN

- *tar xvzf EMAN.src.tgz (or whatever it's called)*
- install any necessary libraries:
 - FFTW (Fourier transforms)
 - GSL (GNU Scientific Library)
 - qt (version 3, usually preinstalled on Linux)
 - OpenGL (and the GLU library, machine dependent)
- *cd eman*
- *./configure --enable-py (or whatever options you like)*
- *make install*

EMAN Source

(subdirectories/files)

- *doc* - EMAN documentation, partially built from sources
- *libEM, libqEM and qscilib* - EMAN Libraries, libEM non-graphical
- *src and local* - 'batch' programs, ie - no GUI
- *boxer, ctfit, eman, glmatrix, helixboxer, qindex, qsegment, v2, v4* - Each GUI program has its own directory
- *python* - eman pythonization support
- *windows* - work in-progress, some level of native Win support
- *chimeraext* - work in-progress, module(s) for chimera
- *configure* - main configuration script, *configure --help*
- **TODO** - Todo list with history of important changes at the end