Refinement at low resolution (lower than ~ 3.5 A resolution)

General remarks

In this section we will discuss the specific settings for the CNS task file `refine.inp` for refinement at low resolution. Compared to refinement at high to medium resolution structures (see the tutorial Refinement at high to medium resolution) it is recommended to:

1. use torsion angle dynamics rather than Cartesian dynamics whenever possible.
2. try refinement without positional minimization, i.e., bond lengths and bond angles should be kept at fixed values.
3. try restrained grouped B-factor refinement as an alternative to standard restrained individual refinement; use the method that produces the lowest Rfree. For restrained grouped B-factor refinement, you need to relax the default target values for B-factor restraints. In this example, the target sigma values for mainchain bonds and angles are set to 15 and 20 A^2, respectively, and those for sidechain bonds and sidechain angles are set to 20 and 25 A^2 respectively. Some experimentation is recommended to determine the optimum values (as indicated by a minimum Rfree value). Note, that it is necessary to explicitly define non-standard B-factor groups, such as ligands (ADP in the particular example below).

However, depending on the resolution and the quality of the diffraction data, restrained individual B-factor refinement and positional (Cartesian) minimization could be beneficial. For example, for the structure of p97/VCP in complex with ADP, we obtained a slightly better Rfree value (about 1%) with restrained individual B-factor refinement and positional minimization. Thus, it is worth trying different refinement strategies and chose the one that produces the lowest Rfree value.

Example

The particular example (p97 in complex with ADP) that we discuss in this tutorial uses torsion angle dynamics and restrained grouped B-factor refinement without positional minimization (the `refine_lowres.inp` file is taken from the `refine.inp` input file, i.e., the only differences are the particular input parameter settings): :

```
cns_solve < refine_lowres.inp > refine_lowres.out
```

Note that this low resolution refinement example uses non-crystallographic symmetry (NCS) restraints as specified by the CNS NCS definition file

`p97_adp_ncs.def`

This example also contains six non-standard ligands, two ADP molecules bound to each of the three protomers. In order for CNS to recognize these non-standard ligands, the topology and parameter files

`p97_adp_cns.top`
`p97_adp_cns.par`

are read.

It is generally advisable to perform many repeats (sometimes referred to as "trials") of the annealing refinement with different seeds that determine the initial random velocities for the molecular dynamics

simulations. For many repeats, it is convenient to setup a C Shell script files that modifies the CNS `refine_lowres.inp` task file. A particular example is:

```
./refine.csh 1
```

Multiple such csh scripts can be run to set up multiple jobs (here we just run it a second time with "./refine.csh 2". This approach of multiple independent jobs is particularly powerful in a distributed computing environment (see below). Generally, the coordinates with the lowest Rfree should be used for further analysis.

A note for structures that are already refined

If extensive annealing is not desired, lower the starting temperature of the slow-cooling schedule from 3000 K to 300 K. In addition, the number of macrocycles should be reduced.

Sorting the best structures obtained from multiple independent refinements

The refined coordinates from multiple independent refinements should be sorted with respect to Rfree. We use a Perl script to do this:

```
./sort.pl refine_*.pdb > sort.list
```

This Perl script creates a new directory "sorted" that contains the final coordinates and the corresponding Fourier coefficient map files (*.hkl files) sorted by Rfree, i.e., sorted/sort_1.pdb is the best coordinate set, the one with the lowest Rfree, and sorted/sorted_1.hkl is the corresponding Fourier coefficient file. The script also produces a convenient summary:

`sort.list`

B-factor sharpening of electron density maps

B-factor sharpening of the electron density maps produced by `refine.inp` can be accomplished in several ways. First, the electron density map files (`refine*.map`) can be read by Coot and then B-factor sharpened using Calculate -> Map Sharpening. This method also works when reading the CNS structure factor file that contains the coefficients for the 2Fo-Fc and Fo-Fc maps (`refine.hkl`) into Coot using File -> Auto OpenMTZ. Second, the CNS structure factor file `refine.hkl` can be read by the CNS `fourier_map.inp` script and B-factor sharpening applied upon computing electron density maps (the coefficients for the Fo-Fc map are in the array called F1, the coefficients for the 2Fo-Fc map are in the array F2).

Averaging electron density maps obtained from multiple independent refinements

The electron density maps resulting form the best structures (i.e., lowest Rfree structures) can be averaged by using the `model_map.inp` script file (see section Refinement at high to medium resolution). This type of averaging can be very powerful to improve model phases (L. M. Rice, P. D. Adams, Y. Shamoo, and A. T. Brunger, Phase improvement by multi-start simulated annealing refinement and structure factor averaging, J. Appl. Crystallography 31, 798-805, 1998). The `model_map.inp` script file also has an option for B-factor

sharpening.

Setup for distributed (grid) computing

It can be very useful to run many jobs independently on different processors. If you have Xgrid installed on your MAC OS X server, here is a script file for the Gridstuffer program:

`refine_xgrid_stuffer.txt`

which uses the unix shell file `refine.csh` described above. A similar approach can be used in any distributed environment, not just Xgrid.

Script to run this tutorial

Back to tutorials Next section
