1) Rosetta and electron density basics

This section provides a brief introduction to using Rosetta, and an overview of using density data within Rosetta.

Overview of Rosetta

The Rosetta documentation is a good source of additional information on several of the tools described in this document. This is available at https://www.rosettacommons.org/docs/latest/Home.

The tools described in this document use the RosettaScripts framework, described at https://www.rosettacommons.org/docs/latest/scripting_documentation/RosettaScripts/RosettaScripts. Briefly, this allows protocols to be defined as a series of atomic "Movers" which manipulate a structure. The format is as follows:

```
<ROSETTASCRIPTS>
    <SCOREFXNS>
    </SCOREFXNS>
    <MOVERS>
    </MOVERS>
    <PROTOCOLS>
    </PROTOCOLS>
    <OUTPUT />
</ROSETTASCRIPTS>
```

Each block contains information in running the protocol: <SCOREFXNS> and <MOVERS> are used to *declare* score functions and movers; while <PROTOCOLS> is where the steps of the protocol are enumerated.

Rosetta tools are run via the command line, with flags controlling general program behavior. Many of the flags specifically for density refinement are outlined in the sections following.

Density scoring in Rosetta

Agreement to density is implemented in Rosetta as an additional energy term. Rosetta assesses agreement to density by computing the density that one would expect to see, given a model, and measuring the agreement of the expected and experimental density.

elec_dens_fast
This scoreterm is recommended for nearly all uses of density refinement in Rosetta. It uses interpolation on a precomputed grid of per-atom scores to approximate the density correlations. This version is significantly faster (~10x) then the exact scoring term below, and is very highly correlated.

These energy terms may be provided to Rosetta in two ways. First, it may be provided in a RosettaScript XML file as input:

```
<Reweight scoretype="elec_dens_fast" weight="35.0"/>
```

For non-Rosetta script applications, the following flag controls the density scoring function weight:

```
-edensity:fast_dens_wt 35.0
```

The recommended weights for each of these terms vary depending on the density map resolution, starting

model quality, and protocol. Section 2 describes how the weights may be tuned. However, the following are good rules of thumb for setting the density weight within Rosetta:

At resolutions better than 2.5Å: an *elec_dens_fast* weight of **65.0** is generally reasonable.
At resolutions between 2.5Å and 3.5Å: an *elec_dens_fast* weight of **50.0** is generally reasonable.
At resolutions worse than 3.5Å: an *elec_dens_fast* weight of **35.0** is generally reasonable.
In centroid mode (described in part 4): an *elec_dens_fast* weight of **10.0** is generally reasonable

At very low resolutions (worse than 6Å), the weight may need to be further reduced. In general, if the Rosetta energies are positive (or significant outliers are flagged by Molprobity or other validation programs) then the weights need to be reduced.

In addition to the score terms above, there are also several flags that control map scoring behavior. Maps are read into Rosetta using either the flag:

```
-edensity::mapfile mapfile.mrc
```

Or from XML:

```
<LoadDensityMap name="loaddens" mapfile="mapfile.mrc"/>
```

Maps may be in either CCP4 or MRC format (the map type is automatically detected from the header info).

The resolution of the map, used when comparing calculated to experimental density, is specified with the flag:

```
-edensity::mapreso 5.0
```

Maps may also be resampled to reduce memory usage and runtime. This is done through the flag:

```
-edensity::grid_spacing 2.0
```

Notice that this flag should *never be more than half the given resolution*, and if using the fast scoring function *never more than a third of the resolution*. For both parameters, **the default is generally fine** (don't resample, and assume the resolution is ~3x the grid sampling).

Finally, one may choose to calculate density using either cryoEM or X-ray scattering factors. At low resolution, this probably makes little difference, but might at resolutions better than about 3.5Å. The default is to use X-ray scattering factors; to turn on cryoEM scattering factors instead, use the following flag:

```
-edensity::cryoem_scatterers
```

Example 1A: Scoring a PDB in Rosetta with density

Most simply, one may wish to simply score a model using Rosetta's energy function including the density terms. This is easily accomplished using the *score_jd2* application. A sample command line to rescore the structure in density is given in *1_rosetta_basics/A_run_rescore.sh*. It illustrates the use of various density flags to provide Rosetta with experimental density information.

```
$ROSETTA3/source/bin/score_jd2.macosclangrelease \
  -database $ROSETTA3/database/ \
  -in::file::s 1isrA.pdb 1issA.pdb \
  -ignore_unrecognized_res \
  -edensity::mapfile 1issA_6A.mrc \
  -edensity::mapreso 5.0 \
  -edensity::grid_spacing 2.0 \
  -edensity::fastdens_wt 35.0 \
  -edensity::cryoem_scatterers \
  -crystal_refine
```

Some flags of note are boldfaced above. First, the input structure is provided with the command -in::file::s. **This is common to many Rosetta applications, and more than one input may be provided; each will be processed independently**. The flags beginning with –edensity:: tell Rosetta about the density map into which it is being fit. The name of the mapfile (in CCP4 or MRC format), the resolution of the map, the grid sampling of the map (*which should never be more than half the resolution*), and the weights on the various fit-to-density scoring functions. These same flags are reused for many different protocols in addition to relax. Finally, the flag -crystal_refine the flag turns on several density-related options related to PDB reading and writing, and should always be used when refining against density data.

Note: The input PDB must be aligned to the density map using some external tool. Rosetta will optionally rigid-body minimize the structure into density before rescoring by providing the flag –edensity::realign min to the application. If this is done, the flag –out::pdb will write the minimized PDB file to a PDB file.

This command line outputs a score file, *score.sc,* that gives, for each structure specified with -in::file::s, the score with respect to each term in Rosetta's energy function. The meaning of individual scoreterms as well as an overview of the Rosetta energy function can be found in the paper:

Alford RF, Leaver-Fay A, Jeliazkov JR, O'Meara MJ, DiMaio FP, Park H, Shapovalov MV, Renfrew PD, Mulligan VK, Kappel K, Labonte JW, Pacella MS, Bonneau R, Bradley P, Dunbrack RL Jr, Das R, Baker D, Kuhlman B, Kortemme T, Gray JJ. The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *J Chem Theory Comput.* 2017 Jun 13;13(6):3031-3048.

Example 1B: Simple refinement into density using RosettaScripts and relax

In this section we introduce RosettaScripts by way of a very simple refinement-into-density example. **RosettaScripts provides an XML scripting interface to Rosetta that allows fine-grained control of protocols.** The syntax is fully described in the Rosetta documentation; however, a very brief introduction is provided here. The basic syntax for the XML is illustrated here (*1_rosetta_basics/B_relax_density.xml*)

```
<ROSETTASCRIPTS>
   <SCOREFXNS>
      <ScoreFunction name="dens" weights="beta_cart">
         <Reweight scoretype="elec_dens_fast" weight="35.0"/>
         <Set scale_sc_dens_byres="R:0.76,K:0.76,E:0.76,D:0.76,M:0.76,
            C:0.81,Q:0.81,H:0.81,N:0.81,T:0.81,S:0.81,Y:0.88,W:0.88,
            A:0.88,F:0.88,P:0.88,I:0.88,L:0.88,V:0.88"/>
      </ScoreFunction>
   </SCOREFXNS>
   <MOVERS>
       <SetupForDensityScoring name="setupdens"/>
       <LoadDensityMap name="loaddens" mapfile="1issA_6A.mrc"/>
       <FastRelax name="relaxcart" scorefxn="dens" repeats="2" cartesian="1"/>
   </MOVERS>
   <PROTOCOLS>
```

```
      <Add mover="setupdens"/>
      <Add mover="loaddens"/>
      <Add mover="relaxcart"/>
   </PROTOCOLS>
   <OUTPUT scorefxn="dens"/>
</ROSETTASCRIPTS>
```

There are three "blocks" of declarations in this script. In the first, <SCOREFXNS> … </SCOREFXNS>, the scorefunctions to be used throughout the protocol are declared; the second, <MOVERS> … </MOVERS>, movers – or atomic operations that modify a structure – are declared; finally, the third, <PROTOCOLS> … </PROTOCOLS>, a full protocol is declared as a sequence of movers.

In this particular example, we declare a single scorefunction, *dens*, which uses the score function *beta_cart* (a default score function, don't need to worry about it), and turns on *elec_dens_fast*, the fit-to-density score, with a weight of 35. We then declare three movers, *SetupForDensityScoring*, *LoadDensityMap*, and *FastRelax*, which sets up the loaded structure for density scoring, loads a map into memory, and then refines the structure using the *FastRelax* protocol. The declared scorefunction, *dens*, is used as an input to the *FastRelax* mover.

To run this script, we use the following command line (*1_rosetta_basics/B_relax_density.sh)*:

```
$ROSETTA3/source/bin/rosetta_scripts.macosclangrelease \
  -database $ROSETTA3/database/ \
  -in::file::s 1isrA.pdb \
  -parser::protocol ex_B1_run_RS_relax_density.xml \
  -ignore_unrecognized_res \
  -edensity::mapreso 5.0 \
  -edensity::cryoem_scatterers \
  -crystal_refine \
  -out::suffix _relax \
  -beta
```

Note: We do not have to specify the density weight or the map file on the command line, since they are handled within the XML file. However, other density options must be specified on the command line. **When using RosettaScripts, the density weights *must* be specified in the XML, the input map may be specified *either way*.**

Finally, in the previous XML file, the tag *cartesian=1* appears, *which refines the structure in Cartesian space*. Rosetta also allows refinement in torsional space, which may be better for capturing domain motion, and for further reduction in model parameters against low-resolution data. *To enable torsional refinement* (*1_rosetta_basics/C_relax_tors_density.xml*), we make three small changes to the XML:

```
<ROSETTASCRIPTS>
   <SCOREFXNS>
      <ScoreFunction name="dens" weights="beta">
         <Reweight scoretype="elec_dens_fast" weight="35.0"/>
         <Set scale_sc_dens_byres="R:0.76,K:0.76,E:0.76,D:0.76,M:0.76,
            C:0.81,Q:0.81,H:0.81,N:0.81,T:0.81,S:0.81,Y:0.88,W:0.88,
            A:0.88,F:0.88,P:0.88,I:0.88,L:0.88,V:0.88"/>
      </ScoreFunction>
   </SCOREFXNS>
   <MOVERS>
       <SetupForDensityScoring name="setupdens"/>
       <LoadDensityMap name="loaddens" mapfile="1issA_6A.mrc"/>
       <FastRelax name="relaxcart" scorefxn="dens" repeats="5" cartesian="0"/>
   </MOVERS>
   <PROTOCOLS>
      <Add mover="setupdens"/>
      <Add mover="loaddens"/>
```

```
      <Add mover="relaxcart"/>
   </PROTOCOLS>
   <OUTPUT scorefxn="dens"/>
</ROSETTASCRIPTS>
```