TUTORIALS

The following are tutorials for using the applications and score terms described in the text of the paper. These tutorials are designed for users who have never used Rosetta before.

Text presented in `fixed width` are commands to be entered into the terminal.

Example files are included in the directory: `tutorial_files`

Tutorial 1: Calculating a Protein Structure's per residue Neighbor Count using *burial_measure_centroid*

This tutorial provides all the information and steps necessary to run the Rosetta application *burial_measure_centroid* to calculate the neighbor counts for all the residues in the crystal structure of myoglobin. The steps presented can easily be used for any protein structure.

1. Prepare your working directory. This directory will contain all of the input and output files that are generated through the course of this tutorial. First create a directory called `tutorial_1` and switch to it:

   ```
   > mkdir tutorial_1
   > cd tutorial_1
   ```

2. Obtain the PDB structure of myoglobin.
   a. Go to http://www.rcsb.org/ and search for the PDB ID 1DWR.
   b. Download the PDB file ("Download Files" -> "PDB Format") and save it as 1dwr.pdb in the directory `tutorial_1`

3. Prepare the PDB for use in Rosetta using the script: clean_pdb.py
   a. The raw PDB file downloaded contains ligands and water molecules. These need to be removed prior to running Rosetta.
   b. Use the script clean_pdb.py (provided with Rosetta) to remove everything except the coordinates marked as "ATOM" in chain A:

   ```
   > python ~/Rosetta/tools/protein_tools/scripts/clean_pdb.py 1dwr.pdb
   A
   ```

   c. Running this script will provide 2 output files: `1dwr_A.pdb` and `1dwr_A.fasta`
   d. In cases of proteins that contain multiple chains, substitute in the chain you wish to analyze in place of the "A" in the above command.

4. Run the application using the following command:

   ```
   > ~/Rosetta/main/source/bin/burial_measure_centroid.linuxgccrelease
   -database ~/Rosetta/main/database -in:file:s 1dwr_A.pdb >
   1dwr_neighbor_count.txt
   ```

   a. Important note: depending upon the operating system you are using and the C++ compiler used when building Rosetta, the application executable might have a different name than what is shown above (please see the Rosetta documentation for more details)

5. After running the application, the result will be a file called `1dwr_neighbor_count.txt`. This file will contain multiple header lines that begin with the string *core*. These can be ignored.

6. The output lines will follow the following format, with a row for each residue in the input structure:

   ```
   G 1 30.3082
   D 2 26.455
   ```

   The first column corresponds to the letter designation for each amino acid, the second column is the residue ID number, and the third column is the calculated neighbor count.

7. If you wish, the numerical parameters used to calculate the neighbor counts can be adjusted from their default values (shown in Equation 2 of the text). These parameters correspond the x-value of the sigmoid functions midpoint (default = 9.0) and the steepness of the curve (default = 0.1). To change these values, include the following flags in the command from step 4 along with the desired values:

   ```
   -dist_midpoint 9.0
   -dist_exponent 0.1
   ```

Tutorial 2: Predicting Protein Structures with *hrf_ms_labeling*

In this tutorial, the steps required to generate *ab initio* models with Rosetta and then rescore them with the score term *hrf_ms_labeling* are presented. For this tutorial, myoglobin is used as an example. The goal is to provide you with the framework needed to be able to run these calculations for any protein you have HRF/HRF labeling data for.

1. Prepare your working directory. This directory will contain all of the input and output files that are generated through the course of this tutorial. First create a directory called tutorial_2 and switch to it:

   ```
   > mkdir tutorial_2
   > cd tutorial_2
   ```

2. Following steps 2-3 from **Tutorial 1**, obtain and prepare the FASTA and PDB for myoglobin (1DWR).

3. Generate the 3mer and 9mer fragment libraries using Robetta
   a. Go to http://robetta.bakerlab.org/ and register (if you are an academic or non-profit user of Rosetta).
   b. Click on the "Submit" link underneath "Fragment Libraries"
   c. Enter in your username or email address into the respective box.
   d. Under "Target Name" enter 1dwr
   e. Copy and paste the text found in 1dwr_A.fasta into the box titled "Paste Fasta" or upload the file 1dwr_A.fasta using the upload feature.
   f. Press "submit"
   g. After your job finishes running (you can check the status by clicking on the "Queue" link under "Fragment Libraries" on the main page), you will receive an email with the results
   h. Download the following files to your directory, tutorial_2: `aat000_03_05.200_v1_3` and `aat000_09_05.200_v1_3`. These files correspond to the 3mer and 9mer libraries respectfully.

i. Rename the two fragment files using the following commands:

```
> mv aat000_03_05.200_v1_3 frags_1dwr_3

> mv aat000_09_05.200_v1_3 frags_1dwr_9
```

j. **Note**: there are other ways to generate the fragment files, please see the Rosetta documentation for more details.

4. Generate an input flags file with the name flags_1dwr_abinitio in the tutorial_2 directory.
   a. This file will contain all of the inputs needed to run the *AbinitioRelax* application.
   Include the following lines in this file (note, the indents are important!):

```
-abinitio
 -relax
-in
 -file
  -fasta 1dwr_A.fasta
  -frag3 frags_1dwr_3
  -frag9 frags_1dwr_9
  -native 1dwr_A.pdb
-out
 -pdb
 -file
  -silent 1dwr_abinitio_silent.out
  -scorefile 1dwr_abinitio_score.sc
-nstruct 10
```

   b. With this set of input flags, the *AbinitioRelax* application will read in the FASTA sequence, fragment libraries, and native structure PDB as inputs. It will output the generated models as PDB files (they will be named S_0000001.pdb, S_0000002.pdb, etc.) along with a silent file and a score file. The total number of structures to generate is set to 10 for this tutorial. Please note that in practice, at least 5,000 models are needed to provide any meaningful results. But for the sake of time, we will only be building 10 models.
   c. Within the score file (`1dwr_abinitio_score.sc`), you will see multiple columns that break down the various score contributions to the total Rosetta score. For this tutorial, the only columns that are important are columns 2

5. Run the *AbinitioRelax* application using the following command:

```
> ~/Rosetta/main/source/bin/AbinitioRelax.linuxgccrelease -database
~/Rosetta/main/database @flags_1dwr_abinitio
```

   a. While the application is running, status information will continually be updated to the screen. If an error occurs, a message will appear in the terminal and the application will stop running.
   b. Generating models will take approximately 1-2 minutes per structure.

6. Generate HRF/HRF labeling data input file.
   a. In order to use the score term *hrf_ms_labeling*, an input file containing the labeled residues along with the natural logarithm of the protection factors is required (please see the text for the definition of a protection factor).

b. Create a new text file with the name `labeling_myoglobin.txt`. This text fill will have the following basic format:

```
# RESIDUE ID AND LNPF FOR MYOGLOBIN
7    4.0943445622
11   4.0989003788
14   4.3705979389
18   2.2462321564
21   3.6018680771
```

c. The first line is a header line describing what is in the file. Each subsequent line contains two columns: the first being the residue ID and the second being the lnPF.

d. **NOTE**: Rosetta will read in your cleaned PDB file and begin the numbering of the residues at 1. So, if your cleaned PDB does NOT begin with residue 1 (i.e. there were missing residues at the start of the sequence), there might be a discrepancy between the residue ID's for your labeled residues in your input labeling file. Be sure to check to make sure that the ID number of the labeled residues correctly corresponds to the numbering of the residues in your PDB (this might require shifting the labeled residue ID numbers).

7. Rescore the *AbinitioRelax* generated PDB models using the *score* application.

   a. Run the following command for each PDB (S_0000001.pdb, S_0000002.pdb, ...):

   ```
   > ~/Rosetta/main/source/bin/score.linuxgccrelease -database
   ~/Rosetta/main/database -in:file:s S_000001.pdb -score:weights
   hrf_ms_labeling.wts -in:file:hrf_ms_labeling
   labeling_myoglobin.txt -centroid_input -out:file:scorefile
   1dwr_rescore_score.sc
   ```

   b. This command will need to be run for each model generated in step 5. For large numbers of models, it is highly recommended to write a script to automate this process.

8. To analyze the results, pull out the following entries from initial output score file (`1dwr_abinitio_score.sc`): *score*, *rms*, *description*. The *score* column represents the raw Ref15 Rosetta score, *rms* is the RMSD to the native structure used as input, and *description* is simply the name of the model. Additionally, extract from the second score file (`1dwr_rescore_score.sc`) the columns labeled *hrf_ms_labeling* and *description*. The *hrf_ms_labeling* refers to the score from just the HRF score term and *description* is the name of the model. To analyze the results, add together the *score* and *hrf_ms_labeling* values for each model to obtain the total score for that given structure. At this point, if a native model was used, a plot can be made of the total score versus RMSD to native.