ROSETTA Comparative Modeling Tutorial – Step- by-step Instructions

*** BLUE text means that these files and/or this information is provided.**
*** RED text means that this material will NOT be conducted during the workshop**
*** Change all environment variables (anything starting with a $) to your local paths**
*** If you want to try making files that already exist (e.g., input files), write them to a new directory!**
(mkdir `$WORKSHOP_ROOT/tutorials/modeling/my_input_model)`

1. Prepare your input files

a. FASTA file of your target sequence:
The `2foxA.fasta` file is already provided for you in the `$WORKSHOP_ROOT/tutorials/modeling/input_model` directory

i. Get sequence in FASTA format from NCBI
- Go to http://www.ncbi.nlm.nih.gov/protein/.
- Type in "2foxA "in the search bar at top.
- Click "FASTA" link to see protein sequence in FASTA format
- Copy all the sequence information, including the line beginning with ">", into a file called `2foxA.fasta`.

ii. Move FASTA file to your input directory

```
mv 2foxA.fasta $WORKSHOP_ROOT/tutorials/modeling/
my_input_model/2foxA.fasta
```

b. Prepare PDB and FASTA files of template structure:
The `1f4pA.pdb` and `1f4pA.fasta` files are already provided for you in the `$WORKSHOP_ROOT/tutorials/modeling/input_model/` directory

i. Download the 1F4P PDB file and FASTA files from http://www.rcsb.org

ii. Clean PDB using the clean_pdb.py script

```
python $ROSETTA_SCRIPTS/clean_pdb.py 1F4P.pdb A
```

iii. Move cleaned PDB file and FASTA file to your input directory

```
mv 1F4P_A.pdb
$WORKSHOP_ROOT/tutorials/modeling/my_input_model/1f4pA.pdb
mv 1F4P.fasta
$WORKSHOP_ROOT/tutorials/modeling/my_input_model/1f4pA.fasta
```

c. 3mer and 9mer fragment libraries:
The `2foxA` fragment files are already provided for you in the `$WORKSHOP_ROOT/tutorials/modeling/input_model` directory (`aa2foxA03_05.200_v1_3` and `aa2foxA09_05.200_v1_3`)

i. Using Robetta (for the purposes of this workshop)
- If you are an academic or non-profit user of ROSETTA, make sure you're registered at http://robetta.bakerlab.org/
- Under "Services," click "submit" under "Fragment Libraries"
- Fill in the form; copy/paste all the text in `2foxA.fasta` into the provided field.
- Under "Target name" put "2foxA" **Note:** If you are benchmarking, would want to exclude homologues.
- Click "Submit." Can see your position in the queue by clicking "Queue" under "Fragment Libraries." This should not take very long.
- Your fragment files should be called `aa2foxA03_05.200_v1_3` and

`aa2foxA09_05.200_v1_3`. Save all the files to `$WORKSHOP_ROOT/tutorials/modeling/my_input_model/`

ii. Using `make_fragments.pl`

- **We will not run `make_fragments.pl` during the workshop!**
- If you are working for a for-profit institution, will need to use the `make_fragments.pl` script in `your_rosetta_directory/rosetta-3.2/rosetta_fragments`.
- In order to use it, will first need to install PSIBLAST, the non-redundant (NR) database, and perhaps PSIPRED
- Will need to modify `make_fragments.pl` in order to reflect the paths specific to your case (will not do during workshop)
- For a usage statement, run: `your_rosetta_directory/rosetta-3.2/rosetta_fragments/make_fragments.pl`

d. Alignment between target and template sequences:

The `2foxA.1f4pA.aln` file is already provided for you in the `$WORKSHOP_ROOT/tutorials/modeling/input_model/`directory

i. Align sequences using CLUSTALW

- Copy/paste sequences for the target (`2foxA.fasta`) and template (`1f4pA.fasta`) proteins, including headers starting with “>”, to the CLUSTALW server at http://align.genome.jp
- Choose “Slow/Accurate” for Pairwise Alignment and “Protein” for sequences
- Copy/paste alignment to a file called `2foxA.1f4pA.aln`
- Remove extra lines that do not contain sequence information
- Reformat lines to “Sequence Name” “Residue Number” “Sequence”:

```
2foxA 1 -MKIVYWSGTGNTEKMAELIAKGIIE
```

ii. Move alignment file to your input directory

```
mv 2foxA.1f4pA.aln
$WORKSHOP_ROOT/tutorials/modeling/my_input_model/2foxA.1f4pA.aln
```

e. PSIPRED secondary structure of target sequence:

The `2foxA.psipred_ss2` file is already provided for you in the `$WORKSHOP_ROOT/tutorials/modeling/input_model/`directory

i. If you ran the Robetta server, you should already have PSIPRED secondary structure prediction file. If not, you can run the PSIPRED server online (see next step).

ii. Perform PSIPRED secondary structure prediction on target sequence

- Copy/paste sequence `2foxA.fasta` to the PSIPRED server at http://bioinf.cs.ucl.ac.uk/psipred/
- Choose “Predict Secondary Structure (PSIPRED v3.0)” and “Mask low complexity regions” (default settings)
- Click link to download results in plain text format

iii. Move the PSIPRED file to your input directory

```
mv psipass2 output
$WORKSHOP_ROOT/tutorials/modeling/my_input_model/2foxA.psipred_ss2
```

f. Options file:

The `comparative_model.options` file is already provided for you in the `$WORKSHOP_ROOT/tutorials/modeling/input_model` directory

- Try to make one on your own. ROSETTA ignores lines beginning with # (these are comments)
- Replace variable names, such as `$WORKSHOP_ROOT` with your specific absolute paths.

```
$ROSETTA_SCRIPTS/replace_env_variables.py
$WORKSHOP_ROOT/tutorials/modeling/input_model/
comparative_model.options
```

- Avoid mixing tabs and spaces. Be consistent in your formatting (tab-delimited or colon-separated; covered by Steven Combs)

2. Run ROSETTA Comparative Model protocol

a. Make sure all the filenames and paths in the options file are correct!

b. Go to the modeling tutorial main directory

c. Type the following command line. It is also found in the `command_lines.txt` file in `$WORKSHOP_ROOT/tutorials/modeling`

```
$ROSETTA_BIN/minirosetta.$ROSETTA_SUFFIX
@$WORKSHOP_ROOT/tutorials/modeling/input_model/comparative_model.options -
database $ROSETTA_DATABASE >&
$WORKSHOP_ROOT/tutorials/modeling/output_model/comparative_model.log &
```

3. Analyze your data

See below for step-by step instructions on clustering your models. See tutorial from Tutorial 1 (De Novo Folding) on “Score and extract PDBs” and “Score vs. RMSD plots” for further instructions on analysis.

ROSETTA Loop Building Tutorial – Step- by-step Instructions

*** BLUE text means that these files and/or this information is provided.**
*** RED text means that this material will NOT be conducted during the workshop**
*** Change all environment variables (anything starting with a $) to your local paths**
*** If you want to try making files that already exist (e.g., input files), write them to a new directory!**
(`mkdir` `$WORKSHOP_ROOT/tutorials/modeling/my_input_loop)`

1. Prepare your input files

a. Loop File:

The `2foxA.loops` file is already provided for you in the `$WORKSHOP_ROOT/tutorials/modeling/input_loop` directory

i. Create a file called `2foxA.loops` in `$WORKSHOP_ROOT/tutorials/modeling/my_input_loop`

ii. Use a visualization tool to help you determine loop start and end residue numbers.

iii. Create one line per loop to be built:

```
LOOP 6 11 0 0 0
```

Column 1	LOOP	The loop file identity tag
Column 2	<integer>	Loop start residue number. NOTE: The starting structure must have real coordinates for all residues outside the loop definition, plus the first and last residue of each loop region.
Column 3	<integer>	Loop end residue number
Column 4	<integer>	Cut point residue number, >=startRes, <=endRes. default - let LoopRebuild choose cutpoint
Column 5	<float>	Skip rate. default - never skip
Column 6	<boolean>	Extend loop. Default false

b. PDB without loop coordinates:

The `2foxA_no_loops.pdb` file is already provided for you in the `$WORKSHOP_ROOT/tutorials/modeling/input_loop` directory

i. Run the remove_loop_coords.py script with your starting model PDB and your loop file:

```
$WORKSHOP_ROOT/py_protein_utils/scripts/remove_loop_coords.py
2foxA.loops 2foxA_start_model.pdb 2foxA_no_loops.pdb
```

ii. Move the PDB file to your input directory

```
mv 2foxA_no_loops.pdb output
$WORKSHOP_ROOT/tutorials/modeling/my_input_loop/2foxA_no_loops.pdb
```

c. 3mer and 9mer fragment libraries (CCD only)

The `2foxA` fragment files are already provided for you in the `$WORKSHOP_ROOT/tutorials/modeling/input_loop directory` `(aa2foxA03_05.200_v1_3` and `aa2foxA09_05.200_v1_3`)

i. Using Robetta (for the purposes of this workshop)

- If you are an academic or non-profit user of ROSETTA, make sure you're registered at http://robetta.bakerlab.org/
- Under "Services," click "submit" under "Fragment Libraries"
- Fill in the form; copy/paste all the text in `2foxA.fasta` into the provided field.

- Under "Target name" put "2foxA" **Note:** If you are benchmarking, would want to exclude homologues.
- Click "Submit." Can see your position in the queue by clicking "Queue" under "Fragment Libraries." This should not take very long.
- Your fragment files should be called `aa2foxA03_05.200_v1_3` and `aa2foxA09_05.200_v1_3`. Save all the files to `$WORKSHOP_ROOT/tutorials/modeling/my_input_loop/`

ii. Using `make_fragments.pl`

- **We will not run `make_fragments.pl` during the workshop!**
- If you are working for a for-profit institution, will need to use the `make_fragments.pl` script in `your_rosetta_directory/rosetta-3.2/rosetta_fragments.`
- In order to use it, will first need to install PSIBLAST, the non-redundant (NR) database, and perhaps PSIPRED
- Will need to modify `make_fragments.pl` in order to reflect the paths specific to your case (will not do during workshop)
- For a usage statement, run: `your_rosetta_directory/rosetta-3.2/rosetta_fragments/make_fragments.pl`

d. Options file:

The `kic.options` and `ccd.options` files are already provided for you in the `$WORKSHOP_ROOT/tutorials/modeling/input_loop` directory

- Try to make one on your own. ROSETTA ignores lines beginning with # (these are comments.)
- Replace variable names, such as `$WORKSHOP_ROOT` with your specific absolute paths.

```
$ROSETTA_SCRIPTS/replace_env_variables.py
$WORKSHOP_ROOT/tutorials/modeling/input_loop/ ccd.options
```

- Avoid mixing tabs and spaces. Be consistent in your formatting (tab-delimited or colon-separated; covered by Steven Combs)

2. Run ROSETTA Loop Building application

a. Make sure all the filenames and paths in the options file are correct!

b. Go to the modeling tutorial main directory

c. Type the following command line. It is also found in the `command_lines.txt` file in `$WORKSHOP_ROOT/tutorials/modeling`

CCD

```
$ROSETTA_BIN/loopmodel.$ROSETTA_SUFFIX
@$WORKSHOP_ROOT/tutorials/modeling/input_loop/ccd.options -database
$ROSETTA_DATABASE >&
$WORKSHOP_ROOT/tutorials/modeling/output_loop/ccd.log &
```

KIC

```
$ROSETTA_BIN/loopmodel.$ROSETTA_SUFFIX
@$WORKSHOP_ROOT/tutorials/modeling/input_loop/kic.options -database
$ROSETTA_DATABASE >& $WORKSHOP_ROOT/tutorials/modeling/output_loop/kic.log
&
```

3. Analyze your data

See below for step-by step instructions on clustering your models. See tutorial from Tutorial 1 (De Novo Folding) on "Score and extract PDBs" and "Score vs. RMSD plots" for further instructions on analysis.

ROSETTA Clustering Tutorial – Step- by-step Instructions

*** BLUE text means that these files and/or this information is provided.**
*** RED text means that this material will NOT be conducted during the workshop**
*** Change all environment variables (anything starting with a $) to your local paths**
*** If you want to try making files that already exist (e.g., input files), write them to a new directory!**
(`mkdir` `$WORKSHOP_ROOT/tutorials/modeling/my_input_cluster)`

1. **Prepare your input files**

   a. Silent Files or list of PDBs:
   The `cluster_all.out` file is already provided for you in the `$WORKSHOP_ROOT/tutorials/modeling/input_cluster` directory

   i. If you ran more than one job, you will need to combine silent files into one file.

   ```
   $ROSETTA_BIN/combine_silent.$ROSETTA_SUFFIX -database
   $ROSETTA_DATABASE -in:file:silent *.out -
   in:file:silent_struct_type binary -in:file:fullatom -out:output
   -out:file:silent cluster_all.out -out:file:silent_struct_type
   binary -out:file:fullatom
   ```

   b. Options file:
   The `cluster.options` file are already provided for you in the `$WORKSHOP_ROOT/tutorials/modeling/input_cluster` directory

   - Try to make one on your own. ROSETTA ignores lines beginning with # (these are comments.)
   - Replace variable names, such as `$WORKSHOP_ROOT` with your specific absolute paths.

   ```
   $ROSETTA_SCRIPTS/replace_env_variables.py
   $WORKSHOP_ROOT/tutorials/modeling/input_cluster/cluster.options
   ```

   - Avoid mixing tabs and spaces. Be consistent in your formatting (tab-delimited or colon-separated; covered by Steven Combs)

2. **Run ROSETTA Clustering application with the clustering.py script**

   i. Run the clustering.py script, which will execute the Rosetta cluster application and output a series of summary files.

   ```
   python $WORKSHOP_ROOT/py_protein_utils/scripts/clustering.py
   --silent=cluster_all.out
   --rosetta=$ROSETTA_BIN/cluster.$ROSETTA_SUFFIX
   --database=$ROSETTA_DATABASE
   --options=cluster.options
   cluster_summary.txt
   cluster_histogram.txt
   ```

3. **Analyze your data**

   i. The `cluster_summary.txt` and other files are provided for you in the `$WORKSHOP_ROOT/tutorials/modeling/output_cluster/` directory

   - If you're not already there, cd into `$WORKSHOP_ROOT/tutorials/modeling/output_cluster`
   - Sort the `cluster_summary.txt` by the score column from lowest -> highest.
     - `sort -rnk4 cluster_summary.txt > cluster_summary_sorted.txt`
   - Take the top 5-10 clusters by size to look at
     - `head -n 10 cluster_summary_sorted.txt`

v. Now you know the tags of the models you want to extract from the binary silent file, which you can do with the following command line:

```
$ROSETTA_BIN/score_jd2.$ROSETTA_SUFFIX -database $ROSETTA_DATABASE
-in:file:silent
$WORKSHOP_ROOT/tutorials/modeling/input_cluster/cluster_all.out
-in:file:silent_struct_type binary -out:output -out:pdb -
out:file:fullatom -in:file:tags S_1F4PA_0410_1 S_1F4PA_0356_1
S_1F4PA_0036 S_1F4PA_0281
S_1F4PA_0127_1 S_1F4PA_0116_1
```