

Comparative Modeling: Single template modeling and loop building

This tutorial will guide you through modeling XP_005597007, a predicted cyclic GMP-AMP synthase in *Equus caballus* (horse). XP_005597007 is homologous to the human protein PDB: 4o68. The crystal structure for 4O68 will be used as the template to guide comparative modeling. Specific regions of low sequence identity will then be remodeled using Rosetta's CCD and KIC loop building protocols.

1. Setup

Comparative modeling requires various input files that are either generated manually or downloaded from the internet. These files have already been created and are available in their appropriate directories but it is recommended that you try to gather/generate these files yourself. **Boldface** indicates specific filenames. *Italics* indicates webpage entries such as query terms or menu selections.

To start, create your own working directory by typing:

```
mkdir my_model
```

Prepared files can be copied from the indicated directories into your working directory at any step if you wish to skip creating a particular file yourself.

a. Target Sequence

With many comparative modeling applications, you will only have your target's amino acid sequence to start with.

2. Prepare the input files

a. Target Sequence

1. Go to <http://www.ncbi.nlm.nih.gov/>
2. Select *protein* from the dropdown box next to the search bar.
3. Type *XP_005597007* in the search bar.
4. Select *FASTA* near the top of page on the left to see the protein sequence in FASTA format.
5. Copy all the sequence information, including the line beginning with ">", into a file called **XP_005597007.fasta**.
6. Copy the XP_005597007.fasta file to a file named **XP_truncated.fasta**
 - Edit XP_truncated.fasta to replace the first line of the file with ">XP0"
 - To ease the modeling process, remove the N- and C- terminal loops:
 - Delete MAASASAADRTEAQSSGLERASSGRRHLALTSRRSTSTPSVAAIGGWARLFGQRAVARWQGSFCGSTSVIFCLL from the beginning of the sequence.
 - Delete GEF from the end of the sequence.

XP_truncated.fasta should look like this:

```
>XP0
ISAPNEFDVMFKLEVPRIELEECNSGAHYFVKFRNPKGNPLSQFLEEEILSASKMLSKFRKIIK
EEIKHVEDTDVIVERKRRGSPAVTLLIRKPKEISVDIILALESKSSWPASTKEGLPINNWLGTKVKNSLR
RQPFYLVPKHAKEGNQFQEETWRLSFSHIEKDILKNHGQSKTCCETHGVKCCRKDCLKLMKYLLQLKKK
FGNRKELDKFCSYHVKTAFHHVCTQDPHDSQWHSNDLESCFDNCVTYFLHCLKTERLEHYFIPGVNLFSSQ
DQIEKISKEFLSKQIEYERNNGYPVF
```

The prepared **XP_truncated.fasta** can be found in the 1_setup/ directory

b. Template structures

The homologous human protein will be used as the template for comparative modeling. This structure is available on the RCSB Protein Data Bank (PDB). The raw structures from the PDB often contain information not necessary for comparative modeling such as attached T4 lysozyme and/or specific ligands. Once a PDB is downloaded for use as a template, this extra information must be removed before it can be used for comparative modeling with RosettaCM.

1. Go to <http://www.rcsb.org>.
2. Search for *4O68*.
3. Click *Download Files -> PDB File (text)*.
4. Remove residues 162-219 from **4O68.pdb** (sequence in **4O68.pdb** begins at residue number 162)
5. Save changes to **4O68_TRUNCATED.pdb**
6. In addition to extra residues, these PDB's contain additional information that is not useful for Rosetta and may cause problems during the modeling. A script has been prepared to remove all of this extraneous information. (Note: `clean_pdb.py` expects to get the PDB filename in all capital letters, without the ".pdb" ending, followed by the chain letter of the chain to extract from the file.)

```
~/rosetta_workshop/rosetta/tools/protein_tools/scripts/clean_pdb.py 4O68_TRUNCATED A
```

This will generate **4O68_TRUNCATED_A.pdb** and **4O68_A.fasta**.

c. Align target sequence to templates

Comparative modeling uses template structures to guide initial placement of target amino acids in three-dimensional space. This is done according to the sequence alignment of target and template. Residues in the target sequence will be assigned the coordinates of those residues they align with in the template structure. Residues in the target sequence that do not have an alignment partner in any template will be filled in during loop building.

1. Go to <http://www.ebi.ac.uk/Tools/msa/clustalo/>
2. Copy/paste all sequence information from your two fasta files (**XP_truncated.fasta** and **4O68_A.fasta** including the ">" header line into clustal.
3. Leave default settings and click *Submit*
4. *Download* the *alignment* to a file called **XP_4O68.aln**

The prepared alignment can be found in the 2_threading/ directory.

d. Fragment files

Loop building will use fragments to remodel loops and fill in missing residues that didn't align with any template residues.

1. Go to <http://robetta.bakerlab.org> and register as an academic or non-profit user.
2. Go to <http://robetta.bakerlab.org/fragmentsubmit.jsp>
3. Fill in your *username*.
4. Put *XP0* under "Target name"
5. Copy/paste all text from **XP_truncated.fasta** into the provided field.
6. Click "*Submit*"
7. See your position in the queue by clicking "*Queue*" under "*Fragment Libraries*". Click on the *job ID* link for your job and refresh to monitor it until completed.
8. While waiting for fragment generation, step 2, threading, can be performed.
9. Once completed, fragment files can be downloaded and should be saved as **XP0_3.frag**s for *aat000_03_05.200_v1_3* and **XP0_9.frag**s for *aat000_09_05.200_v1_3*.

**Prepared fragment files can be found in the 3_loopbuild/ directory.

3. Threading

1. Thread the target sequence over the template PDB using the included script:

```
python2.7 ~/rosetta_workshop/rosetta/tools/protein_tools/scripts/thread_pdb_from_alignment.py \
--template=4068_A --target=XP0 --chain=A --align_format=clustal \
XP_4068.aln 4068_TRUNCATED_A.pdb XP0_on_4068.pdb
```

4. Loop Building

Despite the high sequence identity, certain residues in XP0 could not be aligned to residues in the template. These residues were not assigned any coordinates during the threading process. It is necessary to fill in these missing residues to complete the modeling. In addition to missing residues, loops may also be defined for regions of low identity or important regions where a greater degree of sampling is necessary.

This step requires the following files to be in the same directory:

- **XP0_3.frag**s (downloaded during setup)
- **XP0_9.frag**s (downloaded during setup)
- **XP0_on_4068.pdb** (generated during threading)
- **build_loops.options** (will be created in this step)
- **XP0.loops** (will be created in this step)

a. Define loops: XP0.loops

The loops file tells Rosetta which regions of the protein to remodel using CCD and KIC.

1. Create a file called **XP0.loops**

- Any residues that were not assigned coordinates during the threading have the coordinates 0.000 0.000 0.000 in **XP0_on_4O68.pdb**. View this file to find which residues must be remodeled and included in loop definitions. Additionally, you may wish to include regions of lower identity based on the alignment. The loop file contains one loop region per line. The two residue numbers included for each line must be residues that have previously been assigned coordinates. These residues are the “anchor” residues and will be not remodeled during the loop building process. In other words, if you wish to remodel residues 10-20, you will define the loop region as “9 21.” For the provided alignment, the following loop regions have been defined for remodeling. Depending on your alignment, you may need slightly different loop values.

```
LOOP 34 41
LOOP 93 99
LOOP 181 186
```

A previously generated loops file can be found in the 3_loopbuild/ directory.

b. Create the Options file build_loops.options

The build_loops.options file is already provided for you in the 3_loopbuild/ directory.

When creating an options file, remember these important tips:

- Rosetta ignores comment lines beginning with #.
- Avoid mixing tabs and spaces. Be consistent in your formatting (tab-delimited or colon-separated).

c. Run the Rosetta Loop Modeling application

```
~/rosetta_workshop/rosetta/main/source/bin/loopmodel.default.linuxgccrelease \
@build_loops.options -database ~/rosetta_workshop/rosetta/main/database
```

Note: You will see several “[WARNING] missing heavyatom” messages at the beginning. This is normal and can be ignored.

****This will take a while to run. You may want to open up a different terminal window and start the Rosetta Clustering Tutorial section below, coming back when the run is finished.**

d. Extract PDB files from the silent file

- When the job is finished running, extract the PDB files from the silent files with the following command line.

```
~/rosetta_workshop/rosetta/main/source/bin/score_jd2.default.linuxgccrelease \
-database ~/rosetta_workshop/rosetta/main/database -in:file:silent XP0_on_4O68_loops.out \
-in:file:fullatom -out:pdb
```

5. Analyze your data

See tutorial from De Novo Folding on “Score and extract PDBs” and “Score vs. RMSD plots” for further instructions on analysis.

To Cluster large sets of models, see the Rosetta Clustering tutorial below.

Rosetta Clustering Tutorial

1. Prepare your input files

- Prepare Silent Files or list of PDBs. Since we only generated 5 models previously, we will use a prepared silent file that contains enough models for clustering to be effective:
 - The silent file **XP0_production.out** is already provided for you in the 4_cluster/ directory.
- Prepare Options file:
 - The **cluster.options** file is already provided in the 4_cluster/ directory.
 - Rosetta ignores comment lines beginning with #.
 - Avoid mixing tabs and spaces. Be consistent in your formatting (tab-delimited or colon-separated)

2. Run the Rosetta Clustering application

- Run the clustering.py script, which will execute the Rosetta cluster application and output a series of summary files with names specified on the commandline. (Here “cluster_summary.txt” and “cluster_histogram.txt”)

```
python2.7 ~/rosetta_workshop/rosetta/tools/protein_tools/scripts/clustering.py \  
--rosetta ~/rosetta_workshop/rosetta/main/source/bin/cluster.default.linuxgccrelease \  
--database ~/rosetta_workshop/rosetta/main/database --options cluster.options \  
--silent=XP0_production.out cluster_summary.txt cluster_histogram.txt
```

3. Analyze your data

The **cluster_summary.txt** and associated files are provided in the 4_cluster/ directory.

- Sort the cluster_summary.txt file by the score column from lowest to highest:

```
sort -rnk4 cluster_summary.txt > cluster_summary_sorted.txt
```

- Look at the top 5 clusters by size:

```
head -n 5 cluster_summary_sorted.txt
```

- Extract the models you are interested in viewing from the binary silent file (make sure the silent file XP0_production.out is in the directory you are running from. If not, copy it to the present directory):

```
~/rosetta_workshop/rosetta/main/source/bin/score_jd2.default.linuxgccrelease \  
-database ~/rosetta_workshop/rosetta/main/database -in:file:silent XP0_production.out \  
-in:file:silent_struct_type binary -in:file:fullatom -out:pdb -out:file:fullatom \  
-in:file:tags XP0_on_4068_loopsXP0_on_4068_0450_0001 XP0_on_4068_loopsXP0_on_4068_0492_0001 \  
XP0_on_4068_loopsXP0_on_4068_0497_0001 XP0_on_4068_loopsXP0_on_4068_0386_0001 \  
XP0_on_4068_loopsXP0_on_4068_0352_0001
```

- View the models using protein visualization tool of your choice.
- Examine the PDB files you extracted using any text editor. The overall Rosetta scores for each scoring term as well as the Rosetta scores for each individual residue can be found at the bottom of the model PDB files.