

Antibody Structure Prediction

Bold text means that these files and/or this information is provided

Italicized text means that this material will NOT be conducted during the workshop

Fixed with text means you should type the command into your terminal

If you want to try making files that already exist (e.g., input files), write them to a different directory.

This tutorial assumes that you have Rosetta added to your PATH variable. If you do not already have this done, add the rosetta applications to your path. For the Meilerlab workshop (tcsh shell), do this:

```
setenv PATH ${PATH}:${HOME}/rosetta_workshop/rosetta/main/source/bin
setenv PATH ${PATH}:${HOME}/rosetta_workshop/rosetta/main/source/tools
```

alternatively, for bash shell users:

```
export PATH=${HOME}/rosetta_workshop/rosetta/main/source/bin:$PATH
export PATH=${HOME}/rosetta_workshop/rosetta/main/source/tools:$PATH
```

Rosetta is assumed to be installed at `${HOME}/rosetta_workshop/rosetta` for this tutorial.

Tutorial

This tutorial enables a user to generate a 3D structural model of an antibody variable (Fv) region from its sequence.

The simplest way to create an antibody structure is through the use of the [ROSIE web server](#). On ROSIE, the antibody application uses the input antibody sequence to generate a homology model. The operation is entirely automated, requiring a minimum of user input.

For greater control of the operation the same protocol can be run manually which allows to check intermediate data, intervene with alternative choices, incorporate known experimental data, etc.

1. Create a directory within the RosettaAntibody directory called `my_files` and switch to that directory.

```
mkdir my_files
cd my_files
```

2. Obtain input sequences for modeling. RosettaAntibody needs the sequence of the light chain and the sequence of the heavy chain in FASTA format as an input. There are a couple of things to keep in mind when using RosettaAntibody:

1. Only sequences from the Fv region can be modeled.
2. The six complementary determining regions (CDRs) are determined by conserved cysteine (Cys, C) and tryptophan (Trp, W) residues that identify the location and length of each CDR. Therefore it is imperative to check the input sequences for the inclusion of these residues so that RosettaAntibody will successfully run. This should include Cys residues at position L22, L92, H22, and H92, as well as Trp residues at position L35, H36, and H103 must be present in the input sequences.

We will model the human monoclonal antibody 5J8 that binds and neutralizes a broad spectrum of modern H1N1 influenza viruses including the 2009 pandemic H1N1 virus (Krause et al., J Virol 2011). The antibody was crystallized at a resolution of 1.55 angstroms and the resulting structure was submitted to the Protein Data Bank (PDB) under accession number 4M5Y. The **4m5y_Fv.fasta** file is provided for you in the `input_files` directory. You can copy it:

```
cp ../input_files/4m5y_Fv.fasta .
```

or you can create a text file `4m5y_Fv.fasta` in your current directory in the text editor of your choice and copy the following six lines in that file:

```
>heavy
EVQLVESGPGGLVKPSDILSLTCAVSGYSISSNYYWGWRQPPGKGLEWIGSIYHSGSTYYKP
SLESRLGISVDTSKNQFSLKLSFVSAADTAVYYCARHVRSGYPDTAYYFDKKGQGLTVTS
>light
SYVLTPPPSVSVAPGETARISCGNNIGTKVLHWYQTPGQAPVLVYDDSDRPSGIPERFS
```

GSNSGNTATLTISRVEVGDEADYYCQVWDISTDQAVFGGGTKLTVL

3. Generating a structural model. Generation of a structural model of an antibody from sequence in RosettaAntibody is done using homology modeling techniques; that is, segments from known structures with similar sequences are used. The input sequence is split into several components: light-chain framework (FRL), heavy-chain framework (FRH), CDRs L1-3, H1-3. For each component, RosettaAntibody searches a curated database of known structures for the closest match by sequence and then assembles those structural segments into a model. That model is then used as the input for the next stage.

If you know close related structures and want to use their fragments for the homology modelling you can specify them using the corresponding flags:

```
-antibody:h1_template
-antibody:h2_template
-antibody:h3_template
-antibody:frh_template
-antibody:l1_template
-antibody:l2_template
-antibody:l3_template
-antibody:frl_template
-antibody:light_heavy_template
```

You can also decrease or increase the number of multiple templates to use during grafting using the `-antibody:n_multi_templates` flag

While construction of 10 grafted Fv models (recommended) takes approximately 200 min of CPU time, we will restrict the number of generated models to 1. To make the experiment more realistic we will exclude the 5J8 antibody structures from the templates using `-exclude_pdb` flag.

Type the following command line to run Rosetta's grafting application to find suitable templates, and graft them together to obtain a crude model of the antibody. The following command requires blastp to be installed (e.g. version 2.7.1). Adapt the paths if necessary:

If you are using your own Rosetta copy from github, make sure to execute `git submodule update --init` to get the additional_protocol_data submodule

```
antibody.linuxgccrelease -fasta 4m5y_Fv.fasta \
-antibody::grafting_database \
~/rosetta_workshop/rosetta/database/additional_protocol_data/antibody/ \
-antibody::blastp /dors/meilerlab/apps/Linux2/x86_64/blast/2.7.1/bin/blastp \
-antibody:n_multi_templates 1 -exclude_pdb 4m5y
```

Open the obtained model in Pymol:

```
pymol grafting/model-0.relaxed.pdb
```

and compare it with the crystal structure:

- type "fetch 4m5y" in Pymol's command line
- than type "align" to superimpose the structure and the model

4. (Optional) You might want to assign the CDR loops in your models to the CDR loop clusters described by North et al. and check whether the chosen templates are suitable.

Run the cluster identification application as follows:

```
identify_cdr_clusters.linuxgccrelease \
-s grafting/model-0.relaxed.pdb -out:file:score_only north_clusters.log
```

If you do not want to wait for the Rosetta Antibody job to finish, you can use the following inputs instead:

```
identify_cdr_clusters.linuxgccrelease \
-s ../output_files/grafting/model-0.relaxed.pdb \
-out:file:score_only ../output_files/north_clusters.log
```

North et al. clustered all CDR loop structures by their backbone dihedral angles and named them by CDR type, loop length and cluster size (e.g., 'H1-13-10' is the 10th most common conformation for 13-residue H1 loops). Occasionally, Rosetta chooses templates that are rare or inconsistent with the sequence preferences observed by North et al. For example, if Rosetta recommends the H1-13-10 cluster, the user might also consider the H1-13-1 cluster. Tables 3-7 of North et al. present consensus sequences for each cluster that can inform this decision. Loops and clusters with proline residues are also worth a manual examination. Several clusters of North et al. are contingent on the presence of prolines in particular locations (e.g., L3-9-cis7-1 has a cis-proline at position 7). Because RosettaAntibody relies on BLAST to choose loop templates, occasionally a loop from an uncommon non-cis-proline cluster (e.g., L3-9-2) is chosen. In such cases, it is best to manually select a loop template from the well-populated cis-proline cluster.

5. The grafted models are crude and must be refined, particularly in the CDR H3 loop and the VL-VH orientation. First, the H3 loop is completely remodeled in the context of the antibody framework using the next-generation KIC (NGK) loop modeling protocol. While a large majority (>80%) of known structures of HCDR3 loops have kinked conformation which, however, tends to sample poor by Rosetta, to ensure sampling of the C-terminal kink conformation, atomic constraints are applied. For subsequent high-resolution refinement, the all-atom CDR H3 side chains are recovered, all CDR side chains are repacked, and the CDR side chains and backbones are minimized. The VL and the VH domains are re-docked with a rigid-backbone RosettaDock protocol to remove any clashes created by the new H3 conformation, and the antibody side chains are again repacked. Using NGK, H3 is refined again in the context of the updated VL-VH orientation. The CDRs are packed and minimized again, and the model is saved as a candidate structure. It's recommended to use the first grafted model as the starting point for 1,000 refined models, and the other 9 grafted models as the starting point for 200 refined models each, for a total of 2,800 refined models. It takes around 50 min per model that's why we are going to generate a small number of models.

Copy the set of standard H3 modeling flags **abH3.flags** to your working directory and create a directory for the H3 modeling output:

```
cp ~/rosetta_workshop/rosetta/tools/antibody/abH3.flags .
mkdir H3_modeling
```

Run Rosetta's antibody_H3 application on the model generated during grafting:

```
antibody_H3.linuxgccrelease \
    @abH3.flags -s grafting/model-0.relaxed.pdb -nstruct 1 \
    -auto_generate_h3_kink_constraint -h3_loop_csts_hr \
    -out:file:scorefile H3_modeling_scores.fasc -out:path:pdb H3_modeling/
```

-s specifies the input file **-nstruct** specifies the number of structures generated

If you know that the HCDR3 loop of the modeled antibody has extended conformation skip the following flags:

```
-auto_generate_h3_kink_constraint
-h3_loop_csts_hr
```

Generating the recommended 2,800 antibody structures takes ~2,500 CPU hours. If running 24 processes in parallel on a modern 24-CPU workstation, expect ~4 d of run time. Distributing the work over nodes on a supercomputer can reduce this time to hours.

You can proceed to the next step and analyse pre-generated structures from **output_files/H3_modeling** directory.

6. (Optional) You can check whether the VL-VH orientations of the antibody models are close to the orientations observed in antibody crystal structures found in the PDB. To do this, run the Python script `rosetta/main/source/scripts/python/public/plot_VL_VH_orientational_coordinates/plot_LHOC.py` from within the `output_files` directory using the following command lines:

```
cd ../output_files
python ~/rosetta_workshop/rosetta/main/source/scripts/\
python/public/plot_VL_VH_orientational_coordinates/plot_LHOC.py
```

This script will create a subfolder (lhoc_analysis) with separate plots for each of the four antibody light-heavy orientational coordinate frame (LHOC) metrics:

- Heavy Opening Angle
- Interdomain Distance
- Light Opening Angle
- Packing Angle

Each plot shows the native distribution (gray) and the orientations sampled by Rosetta (black line), as well as the top 10 models (labeled with diamonds) and the 10 different template structures generated during Step 3 (labeled with dots). Antibody models that are outside the native distributions are unlikely to be correct. If all ten low-scoring models are outside the native distribution, consider returning to Step 3 and manually selecting new templates for the relative orientation of the VL and VH chains by using the -antibody:light_heavy_template flag

Note on possible errors:

Running this script requires that you have an environment variable 'ROSETTA' set to the correct path to Rosetta. If you receive the following error when first attempting to run the above Python script:

```
Could not find environment variable ROSETTA.  
Please `export ROSETTA=/path/to/Rosetta`.  
Thanks. Exiting.
```

The provided example command is for bash-style shells. If you're using tcsh (as we are during the workshop), enter the following to set the ROSETTA environment variable:

```
setenv ROSETTA ~/rosetta_workshop/rosetta/  
echo $ROSETTA
```

The last command should print out "/usr/people/sbioguest/rosetta_workshop/rosetta/". For future use, you should point the ROSETTA environment variable to the path to Rosetta you have installed.

Additionally, throughout this protocol, executables are suffixed by the platform and mode for which they were compiled (i.e., in antibody.linuxgccrelease, 'linux' indicates that the antibody executable was compiled on a Linux operating system, 'gcc' indicates the gcc compiler was used, and 'release' indicates it was compiled in release mode). Unfortunately, the use of the .macosclangrelease suffix (MacOS operating system, the Clang compiler, and release mode) is currently hardcoded in the above mentioned script when Rosetta is initially installed. (This will be corrected in future releases.) If you use platform other than MacOS, you will need to make changes so that this Python script can run with your operating system and compiler. Follow the instructions below to include the proper suffixes to use the correct operating system and compiler:

```
cd ~/rosetta_workshop/rosetta/main/source/  
cd scripts/python/public/plot_VL_VH_orientational_coordinates/
```

Using your favorite text editor, replace Line 42 of the file "constants.py" so that it reads:

```
rosetta_LHOC = rosetta_path + '/main/source/bin/packing_angle.linuxgccrelease'
```

The script is also sensitive to the naming scheme of the output files. You will need to change the default suffix provided in Line 27 of the file "ScoreFile.py" to make it work with the generated example files. Using your favorite text editor, replace Line 27, that by default, should contain "self.template_no = decoy_array[name][6]" with:

```
self.template_no = decoy_array[name].split("-")[1][0]
```

The "ScoreFile.py" also includes the plotting conditions used by "plot_LHOC.py", which by default, yields a histogram the same color grey as the plotted template points. We recommend, at a minimum, editing the plotting settings by (again, using your favorite text editor), replacing Line 109 with the following:

```
PDB_hist = plt.hist(self.PDB_angles[coordinate], bins=100, normed=True, \  
                    histtype='step', linewidth=1.0, label='PDB data', color='red')
```

You are welcome to edit the plotting settings as you wish. When you are finished editing the Python script files, make sure to return to the correct working directory to run the "plot_LHOC.py" script:

```
cd ~/rosetta_workshop/tutorials/RosettaAntibody/output_files
```

7. (Optional) Renumbering of antibody models. Standard residue numbering facilitates comparison of different antibodies, but several different numbering schemes are used (such as enhanced Chothia, AHo, IMGT and Kabat). RosettaAntibody uses the Chothia residue numbering scheme by default but provides a conversion application. For example, to convert from Chothia to AHo numbering, run the following command:

```
cd ../my_files/  
antibody_numbering_converter.linuxgccrelease \  
-s ../output_files/H3_modeling/9model-9.relaxed_0020.pdb \  
-input_ab_scheme Chothia -output_ab_scheme AHo
```

8. Rosetta score is an approximation for the free energy, and thus low-scoring models indicate more favorable (better) energies. A subset of the low-scoring models can be selected as a set of final models or as an ensemble for docking or other downstream applications.

To find top-10 best scored models sort the **H3_modeling_scores.fasc** file:

```
cp ../output_files/H3_modeling_scores.fasc .  
sort -nk2 H3_modeling_scores.fasc | head
```

Open the two best models originated from two different templates in Pymol:

```
pymol ../output_files/H3_modeling/9model-9.relaxed_0020.pdb \  
../output_files/H3_modeling/8model-8.relaxed_0006.pdb
```

Still within Pymol, compare them to the crystal structure:

- type "fetch 4m5y" in Pymol's command line
- type "align to" to superimpose the crystal structure and the models

Other criteria to consider while choosing models:

- natural VL-VH orientations falling within the observed distribution
- select models derived from different templates to maintain diversity

Useful References

1. Weitzner BD, Jeliaskov JR, Lyskov S, Marze N, Kuroda D, Frick R, Adolf-Bryfogle J, Biswas N, Dunbrack RL Jr, Gray JJ. Modeling and docking of antibody structures with Rosetta. *Nat Protoc.* 2017 Feb;12(2):401-416. Doi: 10.1038/nprot.2016.180. Epub 2017 Jan 26. PubMed PMID: 28125104; PubMed Central PMCID: PMC5739521
2. Weitzner BD, Gray JJ. Accurate Structure Prediction of CDR H3 Loops Enabled by a Novel Structure-Based C-Terminal Constraint. *J Immunol.* 2017 Jan 1;198(1):505-515. Epub 2016 Nov 21. PubMed PMID: 27872211; PubMed Central PMCID: PMC5173470.
3. Finn JA, Koehler Leman J, Willis JR, Cisneros A 3rd, Crowe JE Jr, Meiler J. Improving Loop Modeling of the Antibody Complementarity-Determining Region 3 Using Knowledge-Based Restraints. *PLoS One.* 2016 May 16;11(5):e0154811. Doi: 10.1371/journal.pone.0154811. eCollection 2016. PubMed PMID: 27182833; PubMed Central PMCID: PMC4868311.
4. North B, Lehmann A, Dunbrack RL Jr. A new clustering of antibody CDR loop conformations. *J Mol Biol.* 2011 Feb 18;406(2):228-56. Doi: 10.1016/j.jmb.2010.10.030. Epub 2010 Oct 28. PubMed PMID: 21035459; PubMed Central PMCID: PMC3065967.
5. Lyskov S, Chou FC, Conchúir SÓ, Der BS, Drew K, Kuroda D, Xu J, Weitzner BD, Renfrew PD, Sri-pakdeevong P, Borgo B, Havranek JJ, Kuhlman B, Kortemme T, Bonneau R, Gray JJ, Das R. Serverification of molecular modeling applications: the Rosetta Online Server that Includes Everyone (ROSIE). *PLoS One.* 2013 May 22;8(5):e63906. doi: 10.1371/journal.pone.0063906. Print 2013. PubMed PMID: 23717507; PubMed Central PMCID: PMC3661552.