# AlgoSB: Modeling RNA-protein interactions

***Before the hands-on session on Tuesday, December 9, install Rosetta and PyMOL.***

**Downloading Rosetta**:
Download Rosetta 3.13: https://downloads.rosettacommons.org/downloads/academic/3.13/
(Free for academic users)
If you are working on a Linux machine, download “Rosetta 3.13 source + binaries for Linux - as one bundle (18G)”
If you are working on a Mac, download “Rosetta 3.13 source + binaries for Mac - as one bundle (14G)”
*These are big files, so the download may take a while.*

**Installing PyMOL**:
The educational license is free and suitable for our purposes. Download here:
https://pymol.org/edu
Follow the installation instructions to install on your computer.

**Hands-on activity for Tuesday, December 9**: Structure prediction of an RNA-protein complex with cryo-electron microscopy data. This exercise will walk you through the process of modeling an RNA structure at a protein interface in a cryo-EM map. The cryo-EM map has been simulated from a high-resolution crystal structure of elongation factor selB bound to selenocysteine insertion sequence (SECIS) RNA (PDB ID: 1WSU). We will explore the impact of various modeling parameters (number of cycles in the Monte Carlo simulation, whether to perform rigid body docking into the map, the docking move size, adding additional cycles of energy minimization).

## Install Rosetta:

1. Extract the zipped Rosetta file that you downloaded above.
2. Add the following lines to your .bashrc (may be .bash_profile on some systems), replacing YOUR_PATH_TO_ROSETTA with your actual path to Rosetta (for example: `/home/src/rosetta/`):

   ```
   export ROSETTA='YOUR_PATH_TO_ROSETTA'
   export RNA_TOOLS=$ROSETTA/main/tools/rna_tools/
   export PATH=$RNA_TOOLS/bin/:$PATH
   export PYTHONPATH=$PYTHONPATH:$RNA_TOOLS/bin/
   ```

3. Type `source ~/.bashrc` (or if you edited `.bash_profile`, then instead type `source ~/.bash_profile`) to activate these paths & tools.
4. Type `python $RNA_TOOLS/sym_link.py`
5. Verify the setup by typing `rna_helix.py -h`. This should print out usage instructions for `rna_helix.py`.
6. Set up the executables for DRRAFTER. Type:

   ```
   ln -s $(ls $ROSETTA/main/source/bin/rna_denovo* | head -1 )
   $ROSETTA/main/source/bin/rna_denovo
   ```

   Then type:

   ```
   ln -s $(ls $ROSETTA/main/source/bin/drrafter_error_estimation* |
   head -1 ) $ROSETTA/main/source/bin/drrafter_error_estimation
   ```

   Then type:

   ```
   ln -s $(ls $ROSETTA/main/source/bin/extract_pdbs* | head -1 )
   $ROSETTA/main/source/bin/extract_pdbs
   ```

7. Add the path to the DRRAFTER script to your `$PATH` (alternatively, you can type the full path to the DRRAFTER.py script each time that you use it). It is found in `$ROSETTA/main/source/src/apps/public/DRRAFTER/`. For example, add the following line to your `.bashrc`:

   ```
   export PATH=$PATH:$ROSETTA/main/source/src/apps/public/DRRAFTER/
   ```

8. Check installation, type `DRRAFTER.py -h`
   This should return the help message for DRRAFTER.py, including all of the command line options.

## Brief explanation of input files:

All of the necessary files for this demo are available in `$ROSETTA/main/demos/public/drrafter/`, where `$ROSETTA` is the path to your Rosetta installation.

`fasta.txt`: The FASTA file listing the full sequence of the complex being modeled. It should contain at least one line that starts with '>' and lists chains and residue numbers for the sequence, e.g. A:136-258 E:1-23. Here we are modeling chain A residues 136-258 and chain E residues 1-23. The subsequent lines should list the full sequence of the complex. Protein residues are specified by uppercase one-letter codes. RNA residues are specified with lowercase one-letter codes ('a', 'u', 'g', and 'c'). Protein residues should be listed before RNA residues.

`secstruct.txt`: A file containing the secondary structure of the complex in dot-bracket notation. Secondary structure for the protein should be specified by dots. The secondary structure should be the same length as the sequence found in the fasta file. For RNA residues, this secondary structure will be enforced during the DRRAFTER run. RNA secondary structures can be predicted computationally with packages such as ViennaRNA. If the secondary structure is not known, it may be necessary to test several different secondary structures in separate DRRAFTER jobs (or ideally the secondary structure would be determined through biochemical experiments).

`1wsu_simulated_7A.mrc`: The density map file in mrc format (ccp4 is also acceptable). For this demo, this map has been simulated from PDB ID 1WSU at 7Å.

`RNA_helix.pdb`: This is an ideal RNA helix corresponding to residues E:1-4 and E:20-23. Ideal RNA helices can be generated using rna_helix.py (documentation on the RNA tools page). This PDB was generated with the following command:

```
rna_helix.py -seq ggcg cgcc -o RNA_helix.pdb -resnum E:1-4 E:20-23 -
extension static.linuxgccrelease
```

Note that you may need to change the extension in the above command (see instructions in the RNA tools documentation). PDBs used in DRRAFTER must have chain IDs (i.e. you cannot provide -resnum 1-4 20-23 to rna_helix.py to generate an RNA helix for DRRAFTER).
protein_fit_into_density.pdb: The protein structure that has been fit into the density map. For this demo, this is a crystal structure of the unbound protein (PDB ID 1LVA). The sequence of this protein structure should exactly match the sequence in the fasta file.

`protein_and_RNA_helix_fit_into_density.pdb`: Coordinates of both the protein structure and the RNA helix (from RNA_helix.pdb) fit into the density map. This will be the starting conformation for the DRRAFTER run. (The relative positions of the protein and RNA helix will be allowed to change though.) The order of the residues in this PDB file should match the order of the residues in the fasta file.

## Running DRRAFTER:

1. Create and change into a new directory: `DRRAFTER_demo/` (`mkdir DRRAFTER_demo` then `cd DRRAFTER_demo`)
2. Copy the demo files into your `DRRAFTER_demo` directory: `cp -r $ROSETTA/main/demos/public/drrafter/* .`

3. Visualize the demo files in PyMOL, especially `1wsu_simulated_7A.mrc` and `protein_and_RNA_helix_fit_into_density.pdb`
4. Use `DRRAFTER.py` to set up the run. Type:

```
DRRAFTER.py -fasta fasta.txt -secstruct secstruct.txt -start_struct
protein_and_RNA_helix_fit_into_density.pdb -map_file
1wsu_simulated_7A.mrc -map_reso 7.0 -residues_to_model E:1-23 -
include_as_rigid_body_structures protein_fit_into_density.pdb
RNA_helix.pdb -absolute_coordinates_rigid_body_structure
protein_fit_into_density.pdb -job_name demo_run -dock_into_density -
demo_settings -rosetta_directory $ROSETTA/main/source/bin/
```

Note that `-rosetta_directory` needs to provide the path to your Rosetta executables (if you installed DRRAFTER following the instructions above, then `$ROSETTA/main/source/bin/` should be fine). If the path to the Rosetta executables is already in your system `PATH`, then you can omit the `-rosetta_directory` flag.

Note also the `-demo_settings` flag: this flag is designed to make the DRRAFTER run finish quickly, and should not be used for normal runs. Specifically, `-demo_settings` is equivalent to the following options: `-cycles 500 -extra_flags 'rnp_high_res_cycles 0' 'minimize_rounds 1' 'no_filters' 'nstruct 10'`.
For normal runs: the number of structures built per DRRAFTER job can be set with e.g. `-extra_flags 'nstruct 2000'`.

This will create the following output files:

`fasta_demo_run.txt`: The FASTA file for the region that will be included in the DRRAFTER run.

`secstruct_demo_run.txt`: The file specifying the secondary structure for the region that will be included in the DRRAFTER run.

`coord_csts_demo_run.txt`: A file describing coordinate restraints that will be applied during the DRRAFTER run. By default, residues in the starting structure will be restrained to be within 10 Å of their initial coordinates. This can be turned off with -no_csts and the distance at which the restraints will be activated (default 10 Å) can be controlled with -cst_dist.

`flags_demo_run`: A file listing all of the Rosetta options for the run.

`init_struct_demo_run.pdb`: The starting structure for the DRRAFTER run. In this case, this structure is identical to protein_and_RNA_helix_fit_into_density.pdb.

`DRRAFTER_command`: This file contains the command to run the DRRAFTER job.

3. Run the `DRRAFTER_command`. This can be done either by typing:

```
source ./DRRAFTER_command
```

OR by copying the line in the DRRAFTER_command file to the command line:

```
/your/path/to/rosetta/executables/rna_denovo @flags_demo_run
```

This will take several minutes to run and it should create a file called `demo_run.out`, which contains all of the structures from the run.

4. Extract PDB files from the compressed output file created in the previous step. Type:

```
extract_lowscore_decoys.py demo_run.out 10
```

This extracts the 10 best scoring structures from the run and will create 10 PDB files named demo_run.out.1.pdb, demo_run.out.2.pdb, etc. Note that in this case we only generated 10 structures total, but for a real run it is recommended that you build at least 2000-3000 structures. This can be done by setting the number of structures built per DRRAFTER run with -extra_flags 'nstruct 2000' (if this option isn't provided and the -demo_settings option isn't supplied, then default=500).

5. Look at the structures! Open them in PyMOL. You will see that all of the missing RNA residues have been built. The protein and RNA helix have also moved slightly from their initial positions.

6. Estimate the error in the DRRAFTER models. Again, use the DRRAFTER.py script to do this. Type:

```
DRRAFTER.py -final_structures demo_run.out.1.pdb demo_run.out.2.pdb
demo_run.out.3.pdb demo_run.out.4.pdb demo_run.out.5.pdb
demo_run.out.6.pdb demo_run.out.7.pdb demo_run.out.8.pdb
demo_run.out.9.pdb demo_run.out.10.pdb -estimate_error -
rosetta_directory $ROSETTA/main/source/bin/
```

This will print out information about the error estimation to your screen. For example (the actual numbers may vary!):

```
apps.public.DRRAFTER.drrafter_error_estimation: ###########################################
apps.public.DRRAFTER.drrafter_error_estimation: Mean pairwise RMSD (convergence): 5.41042
apps.public.DRRAFTER.drrafter_error_estimation: Estimated minimum RMSD: 3.65777
apps.public.DRRAFTER.drrafter_error_estimation: Estimated mean RMSD: 5.39297
apps.public.DRRAFTER.drrafter_error_estimation: Estimated RMSD of median structure: 4.73551
apps.public.DRRAFTER.drrafter_error_estimation: Median structure: demo_run.out.1.pdb
apps.public.DRRAFTER.drrafter_error_estimation: ###########################################
```

All numbers have units of Å. The mean pairwise RMSD describes the "convergence" of the run, i.e. how similar the final structures are to each other. The estimated RMSD (root mean square deviation) values to the "true" coordinates are based on this convergence value. The estimated minimum RMSD predicts the best accuracy of the final structures. The estimated mean RMSD predicts the average RMSD accuracy of the final structures. The median structure is determined to be the final structure with the lowest average pairwise RMSD to the other final structures. The accuracy estimate of this model is also printed to the screen.

For reference, example output is provided in the `example_output/` directory.

7. How do the options above impact the models that are built? Try the following modifications and see how they impact how the final models and error estimates change:

- Try removing the `-dock_into_density` flag. This will turn off docking moves for all of the structures listed for `-include_as_rigid_body_structures`
- Try running with more cycles in the Monte Carlo simulation: Replace `-demo_settings` with `-cycles **1000** -extra_flags 'rnp_high_res_cycles 0' 'minimize_rounds 1' 'no_filters' 'nstruct 10'`
- Add a second round of minimization: Replace -`demo_settings` with `-cycles 500 -extra_flags 'rnp_high_res_cycles 0' 'minimize_rounds 2' 'no_filters' 'nstruct 10'`
- Increase the docking move size: `-docking_move_size 1.0`

## Additional information

See the DRRAFTER documentation here:
https://www.rosettacommons.org/docs/latest/application_documentation/rna/drrafter