Peptide Design

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 width 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! (mkdir my_dir)

In addition to following this sample peptide design problem, the user is encouraged to review the Rosetta user guide including the section on peptide design and docking movers for use with RosettaScripts.

https://www.rosettacommons.org/docs/latest/

Peptide Design using PeptiDerive Server

For the purposes of this exercise we will model a protein/protein complex from a published structure, Eph receptor bound to ephrin (PDB: 3GXU).

1. Navigate to the peptide design directory where you will find the **step_1_peptiderive/**, and **step_2_flexpepdock/** folders. Move into the peptiderive **input/** folder.
2. Download protein complex from PDB
   - The *clean_pdb.py* script will allow you to automatically download a PDB file and clean it of information other than the desired protein coordinates. The 'A' option tells the script to obtain chain A only. The full crystal structure consists of two monomers that must be cleaned separately.

     ```
     ~/Rosetta/tools/protein_tools/scripts/clean_pdb.py 3gxu A
     ~/Rosetta/tools/protein_tools/scripts/clean_pdb.py 3gxu B
     ```

   - Combine the cleaned monomers into their complex.

     ```
     cp 3gxu_A.pdb complex.pdb
     cat 3gxu_B.pdb >> complex.pdb
     ```

3. Upload complex structure to PeptiDerive Server. https://rosie.rosettacommons.org/peptiderive/submit
   - Submit pdb by clicking **Choose File** next to "Input PDB file"
   - Provide a brief description if desired
   - If you do not specify receptor and partner then PeptiDerive will sample both
   - Change peptide length if desired.
   - **Suggested**: Provide an email address so that you will be notified when your job is done.
   - When job is done download file as tar archive
   - Move downloaded file into output directory. To extract files use command:

     ```
     tar -xvf (file_name)
     ```

   - Important output files (to be used as input for next step)
     receptorA_partnerB_10aa_best_linear_linear_peptide_complex.pdb
     receptorB_partnerA_10aa_best_linear_linear_peptide_complex.pdb

PeptiDerive without Server

- Navigate to **commands_if_not_using_server/** subdirectory
- Here we are using Rosetta Scripts to run the peptiderive mover. Everything is setup. You just need to point the command line to your version of Rosetta, the directory, and your input file:

  ```
  ~/rosetta_workshop/rosetta/main/source/bin/rosetta_scripts.linuxgccrelease \
  @ options.txt -in:file:s ../input/complex.pdb
  ```

- Important output files (to be used as input for next step)

  receptorA_partnerB_10aa_best_linear_linear_peptide_complex.pdb

  receptorB_partnerA_10aa_best_linear_linear_peptide_complex.pdb

Redock the Peptide with FlexPepDock

We want to redock the peptide from the previous step to ensure it stays in place without the context of the larger protein it was derived from. If the peptide moves significantly, it is unlikely to compete with the native protein-protein complex.

1. Change to the **step_2_flexpepdock/** directory
2. Copy outputs from PeptideDerive into input subdirectory.

   ```
   cp ../step_1_peptiderive/output/receptorA_partnerB_10aa_best_linear_linear_peptide_complex.pdb \
     server_input_receptor_A/.

   cp ../step_1_peptiderive/output/receptorB_partnerA_10aa_best_linear_linear_peptide_complex.pdb \
     server_input_receptor_B/.
   ```

3. Upload complexes to FlexPepDock server http://flexpepdock.furmanlab.cs.huji.ac.il/
   - Input your PDB by clicking on **Choose File** button
   - Add your email address as the docking can take some time
4. When docking is complete, download output files

   We want a peptide that remains stable in the binding interface and has a sharp energy funnel.

FlexPepDock without Server

1. Change to sub-directory **without_server/**
2. Copy protein-peptide complex to here.
3. Prepack the initial complex

   The pre-packing mode optimizes the side-chains of each monomer according to the Rosetta energy function. Unless you know what you are doing, we strongly recommend pre-packing the input structures, and applying one of the peptide docking protocols to the resulting pre-packed structures, as this can improve model selection considerably (see below). However, in cases where side-chains have been previously optimized by Rosetta using the same scoring function, this step can be skipped.

   ```
   ~/rosetta_workshop/rosetta/main/source/bin/FlexPepDocking.linuxgccrelease \
      -in:file:s receptorA_partnerB_10aa_best_linear_linear_peptide_complex.pdb \
      -flexpep_prepack -ex1 -ex2aro
   ```

4. Dock Peptide

   Docking of 100 peptides should be sufficient to tell if peptide is stable at interface. However, you are free to choose a higher density sampling. Because of the speed of this protocol, 10,000 decoys is not difficult to model. (Though for time, the example command below only does 10.)

   ```
   ~/rosetta_workshop/rosetta/main/source/bin/FlexPepDocking.linuxgccrelease \
     -in:file:s receptorA_partnerB_10aa_best_linear_linear_peptide_complex_0001.pdb \
     -pep_refine -ex1 -ex2aro -use_input_sc \
     -native receptorA_partnerB_10aa_best_linear_linear_peptide_complex.pdb \
     -nstruct 10 -out:path:all output_files/
   ```

5. Generate energy funnel and analyze top 10 models.