Motif Grafting onto Protein Scaffolds

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`)

Side Chain Grafting Tutorial This tutorial is designed to graft functional motifs onto protein scaffolds. There are two different methods that are provided within this tutorial: *Side Chain Grafting* and *Backbone Grafting*. A library of protein scaffolds is computationally scanned for possible graft sites. If the motif and scaffold backbones superimpose with very low root mean squared deviation (RMSD<0.5), then only hot spot side chains need be transplanted from the motif to the corresponding positions in the matching site of the scaffold. This is known as Side Chain Grafting. Subsequently, surrounding residues on the scaffold surface that are in contact with the target are designed for favorable interactions.

Side Chain Grafting makes the minimal number of changes to the scaffold, increasing the chances of obtaining correctly folded designs during experimental validation. However, often side chain grafting is not possible because the motif and scaffold structures are too dissimilar. In these cases, even though the motif and scaffold may have very different structures, it is still possible to use an alternative method known as Backbone Grafting. During Backbone Grafting, the algorithm looks for segments of the scaffold backbone that align closely to the termini of the motif (both N- and C-terminal sides), and then the scaffold segment between these alignment points is replaced by the motif. This technique is extremely versatile, for example, a loop in the scaffold might be replaced by a peptide motif with different secondary structure, or even with a different amino acid length. Since the changes to the scaffold structure following Backbone Grafting can disrupt the overall fold, it is important to design the hydrophobic core to support the new backbone structure of the scaffold, followed by design of the protein-protein interface. The Backbone Grafting procedure often introduces many mutations to the scaffold, requiring careful filtering of designs to select those that present quality interfaces and high stability of the new scaffold.

In this tutorial we are going to use a co-crystal structure of Estrogen Receptor in complex with a helical peptide from a transcriptional co-activator (1GWQ.pdb). We will follow the steps described below to design a protein binder for ER.

1. Definition of the binding motif for interface design
2. Preparing a scaffold database
3. Matching for putative scaffolds (i.e., motif grafting)
4. Sequence design
5. Selection and Improvement of Designs

The first two steps are similar for both Side Chain and Backbone Grafting. Step 3 will be different because the two methods use different algorithms to search for putative scaffolds to graft the binding/functional motif. Last two steps are also similar but more rigorous for Backbone Grafting.

Create a directory in the SideChainGraft directory called my_files and switch to that directory. Although many files you need for the tutorial are located in the input_files directory, we will work from my_files for the rest of this section of the tutorial.

```
cd ~/rosetta_workshop/tutorials/scaffolding/SideChainGraft/
mkdir my_files
cd my_files
```

A. Prepare the input files for motif grafting.

In this step we will define the binding/functional motif that we want to graft onto a protein scaffold.

1. Download the co-complex from the Protein Databank (PDB). This complex is under the PDB ID "1gwq".

The 1GWQ.pdb file is provided in the input_files directory. However the instructions for downloading this PDB file are also provided below.

1. Go to http://www.rcsb.org and type 1gwq in the search bar.
2. Click on "Download Files" on the right side of the page, then "PDB Format".
3. Save the PDB file in the my_files directory as 1GWQ.pdb.
4. Prepare the PDBs for running through Rosetta.

In general before running a PDB through Rosetta you should remove water molecules and all ligands that are non-essential to your protocol. We will use an automated script to do this processing.

The 1GWQ.pdb file contains a dimer of ER-alpha bound to helical peptides. We want to pull the target structure, i.e., ER-alpha (chain A, renamed as context.pdb) and the binding motif structure (chain C, renamed as motif.pdb) from the PDB 1GWQ.

```
python2.7 ~/rosetta_workshop/rosetta/tools/protein_tools/scripts/clean_pdb.py 1GWQ A
python2.7 ~/rosetta_workshop/rosetta/tools/protein_tools/scripts/clean_pdb.py 1GWQ C
mv 1gwq_A.pdb context.pdb
mv 1gwq_C.pdb  motif.pdb
```

Copy context.pdb and motif.pdb to my_files/ directory.

```
cp ../input_files/context.pdb .
cp ../input_files/motif.pdb .
```

B. Preparing the Scaffold Database

To prepare an inclusive scaffold database that can be searched for a variety of structural motifs, you can download structures from the PDB (http://www.rcsb.org) based on the following four criteria using advanced search module: (1) crystal structures with high-resolution x-ray diffraction data (< 2.5 A), (2) the proteins had been reported to be expressable in E. coli(this simplifies later experimental characterization), (3) a single protein chain in the asymmetric unit (MotifGraft only works with monomeric scaffolds as grafting targets), and (4) no bound ligands or modified residues.

In some circumstances, a focused scaffold library may produce more useful matches. For our particular example, the peptide that seeds interface design has an alpha-helical conformation. Therefore, we also prepared a small focused scaffold library of 18 helical proteins.

A scaffold database is provided in the scaffolds/ directory.

The scaffold PDB files were formatted for ROSETTA and subjected to an energy minimization step as described below.

1. *For this tutorial, skip this step.* Make a list of all pdb files being used as scaffold.

   ```
   ls   *.pdb > pdb_files.list
   ```

2. *For this tutorial, skip this step.* Relax the pdb structures while constraining the structure to its initial coordinates. Note that the "\" at the end of line 1 only marks the end of the line. If you copy-paste the command into the terminal, remove the "\" before running so that the command is one line.

   ```
   ~/rosetta_workshop/rosetta/main/source/bin/relax.linuxgccrelease -ignore_unrecognized_res \
       -relax:constrain_relax_to_start_coords -ex1 -ex2 -use_input_sc  -l  pdb_files.list
   ```

3. Create a list containing all relaxed PDB files within the scaffold database, which are located in the ../scaffolds/ directory.

```
ls   ../scaffolds/*.pdb > scaffolds.list
```

C. Motif Grafting and Sequence Design

Motif matching and interface design are distinct conceptual steps, but due to the flexibility of the RosettaScripts framework, both can be included in a single computational step.

Since putative scaffold matching is different for Side Chain and Backbone Grafting, we are first going to use Side Chain grafting procedure. It is recommended that you attempt Side Chain Grafting before Backbone Grafting for your own functional motif, as it requires less changes in the protein scaffold, increasing the chances of obtaining correctly folded designs during experimental validation.

Now that we have our input pdb files for the motif and context ready along with the scaffolds database to scan and put the motif onto putative scaffolds, we will perform Side Chain Grafting using the MotifGraft_sc.xml script.

Copy the MotifGraft_sc.xml script from scripts/ directory to my_files/ directory.

```
cp ../scripts/MotifGraft_sc.xml .
```

Execute the MotifGraft_sc.xml rosettascripts using the following command. It will take approximately 15 minutes to generate 1 scaffold for each of the 18 input protein strutcures. Again, if you copy-paste the command, make sure to remove any "\" at the end of each line so that the command is one line.

```
~/rosetta_workshop/rosetta/main/source/bin/rosetta_scripts.linuxgccrelease \
    -l scaffolds.list  -use_input_sc  -ex1  -ex2  -nstruct 1 \
    -parser:protocol MotifGraft_sc.xml
```

Execution of this script will generate one design for each scaffold. To generate more than one design, you will need to use the MultiplePoseMover. See https://www.rosettacommons.org/docs/latest/scripting_documentation/RosettaScripts/Movers/movers_pages/RosettaScripts-MultiplePoseMover for documentation.

The expected_output/ directory has designs from a previous run. One should look at the designs in Pymol.

For further explanation of the options used in the XML script, see this methods paper http://www.ncbi.nlm.nih.gov/pubmed/27094298.

D. Selection and Improvement of Designs

To date, no computational method has been developed that can predict with perfect accuracy which designs will be functional when challenged experimentally. Therefore, it is wise to proceed with designed sequences that present good metrics by multiple criteria.

1. Designs are initially filtered based on calculated metrics for interface quality, including a favorable binding energy (ddG < 0 ROSETTA energy units, ideally the energy should be lower than the native interface from which the motif was taken), high shape complementarity (Sc >0.65), and a low number of buried unsatisfied hydrogen-bonding atoms. In the XML scripts above, these filters report to a score file and will also be appended at the end of any ROSETTA output PDBs. If you are generating more than one designs per scaffold (for example, 10 designs per scaffold), you can select them based on the total score before looking for ddG, Sc and other metrics.

2. Once a set of designs has been selected based on the calculated metrics, it is important to perform human-guided inspection of the designed structures. There are many qualities of interfaces that are apparent to structural biologists that are not captured in standard metrics. Two common defects in ROSETTA-designed structures that are very important to avoid are: *i*) buried charged residues and *ii*) under-packed interfaces dominated by alanine residues.

3. Reverting designs back to native residues: It is also important to consider whether the designed scaffold will fold to its intended structure; having a spectacular interface on a computational model is irrelevant if the protein cannot fold in an experimental setting. This is particularly problematic for designed interfaces that have a large surface area dominated by hydrophobic residues. It is generally assumed that the probability of a designed sequence properly folding is inversely correlated with the number of mutations imposed on the scaffold during the design process. Therefore, it is beneficial to be conservative and make as few mutations as possible by reverting residues back to their native identities in a post-design stage. For this step, we will revert one of the scaffold designs 1ji6_0001_0001.pdb to its native sequence from 1ji6_0001.pdb in complex with the target (context.pdb).

```
cat context.pdb ../scaffolds/1ji6_0001.pdb >nativecplx.pdb
```

Note that ../scaffolds/1ji6_0001.pdb here is the protein structure that was designed during Side Chain grafting by putting the motif.pdb onto a protein structure to generate 1ji6_0001_0001.pdb.

```
~/rosetta_workshop/rosetta/main/source/bin/revert_design_to_native.linuxgccrelease \
    -revert_app:wt  nativecplx.pdb -revert_app:design 1ji6_0001_0001.pdb -ex1 -ex2 -use_input_sc
```

4. Manually adjusting designs: The user may wish to correct a number of frequent problematic features in ROSETTA designs, such as hydrophobic residues at the water-exposed interface edge, revert designed residues back to their native identities, mutate buried charged residues to hydrophobics, etc. There are no hard rules for manually improving designs; it is simply a matter of the designers preference and experience.
5. Filtering Designs based on folding probability: Many designed sequences will not fold correctly when experimentally tested. We have found structure prediction to be a powerful filter; the designed amino acid sequences when subjected to structure prediction calculations should yield similar structures to the designed models. If structure prediction returns an alternative conformation, or fails to converge on an energy minimum in a conformational landscape, then it is unlikely that the designed sequence will correctly fold.

Backbone Grafting Tutorial All the steps in Backbone Grafting tutorial are same as in the Side Chain Grafting Tutorial. We have changed the scaffolds database for this part. The MotifGraft_bb.xml script incorporates the backbone grafting algorithm for scaffold matching.

1. Create my_files directory in BackboneGraft directory.

```
cd ../../BackboneGraft/
mkdir my_files
cd my_files/
```

2. Motif and Context pdb files

```
cp ../input_files/motif.pdb .
cp ../input_files/context.pdb .
```

3. Make Scaffolds list.

```
ls  ../scaffolds/*.pdb > scaffolds.list
```

4. Motif Grafting and Sequence Design

```
cp ../scripts/MotifGraft_bb.xml .
```

and execute the backbone grafting script using following commandline.

```
~/rosetta_workshop/rosetta/main/source/bin/rosetta_scripts.linuxgccrelease \
    -l scaffolds.list -use_input_sc -nstruct 1 -parser:protocol MotifGraft_bb.xml
```

Running MotifGraft_bb.xml takes longer than MotifGraft_sc.xml.

5. Selection and Improvement of Designs

Designs from backbone grafting require extra attention, as the engineering of a protein core to support the grafted motif can be challenging. Therefore, one should check to see how motif placement has changed the structure of initial scaffold using PyMol or other protein visualization tool.