ROSETTA PROTEIN-PROTEIN DOCKING TUTORIAL

STEP-BY-STEP

1. Preparing the input file.
    a. Go to http://www.rcsb.org/pdb/explore.do?structureId=2VYR and download the pdb file. Optionally if your copy of pymol has a pdb loader service load of the Antibody-Antigen Complex 2VYR.
    b. Once the complex has loaded, select A -> remove waters.
    c. From the command line type "select complex, chain B chain C and chain G"
    d. Go to file -> save molecule -> complex. Save this file in $WORKSHOP_ROOT/tutorials/protein-proteindocking/tutorial_files/complex.pdb
2. Preparing file for docking.
    a. First change chains B and C's to chain A, to be viewed as one entity.
    b. cd $WORKSHOP_ROOT/tutorials/protein-proteindocking/tutorial_files/
    c. python $ROSETTA_SCRIPTS/set_pdb_chain_id.py --original_chain=B --new_chain=A complex.pdb complex_b_to_A.pdb
    d. python $ROSETTA_SCRIPTS/set_pdb_chain_id.py --orginal_chain=C --new_chain=A complex_b_to_A.pdb complex_c_to_A.pdb
    e. Next, renumber the PDB so the starting amino acids start at 1.
    f. python $ROSETTA_SCRIPTS/clean_pdb.py complex_c_to_A.pdb ignorechain
    g. Change the final output, to an easy to remember input.pdb
    h. mv complex_c_to_A_ignorechain.pdb input_pdb.pdb
3. Preparing the XML file for Rosetta Scripts. Using Rosetta Scripts, we can mimic the protein-protein docking protocol.
    a. The skeleton structure XLM script looks as follows.

```
<dock_design>

          <SCOREFXNS

          </SCOREFXNS>

        <TASKOPERATIONS>

        </TASKOPERATIONS>

        <FILTERS>

        </FILTERS>
        <MOVERS>
```

```
        </MOVERS>
        <APPLY_TO_POSE>
        </APPLY_TO_POSE>
        <PROTOCOLS>
        </PROTOCOLS>
    </dock_design>
```

b. For this tutorial we will only use task of operations and movers.
c. Open a new text file and add the skeleton xml file , save as : $WORKSHOP_ROOT/tutorials/protein-proteindocking/tutorial_files/docking.xml
d. In your favorite text editor add these movers.

```
<dock_design>
        <SCOREFXNS>
        </SCOREFXNS>
     <TASKOPERATIONS>
                <InitializeFromCommandline name=ifcl/>
         </TASKOPERATIONS>

        <FILTERS>
        </FILTERS>
        <MOVERS>
         <Docking name=dock_low score_low=score_docking_low score_high=score12
fullatom=0 local_refine=0 optimize_fold_tree=1 conserve_foldtree=1 design=0
task_operations=ifcl/>
         <Docking name=dock_high score_low=score_docking_low score_high=score12
fullatom=1 local_refine=1 optimize_fold_tree=1 conserve_foldtree=1 design=0
task_operations=ifcl/>
         <PackRotamersMover name=pr scorefxn=score12/>
                <MinMover name=min scorefxn=score12 chi=1 bb=1 jump=1
tolerance=0.01/>

    </MOVERS>
        <APPLY_TO_POSE>
        </APPLY_TO_POSE>
        <PROTOCOLS>
       <Add mover_name=dock_low/>
       <Add mover_name=pr/>
       <Add mover_name=min/>
       <Add mover_name=dock_high/>
       <Add mover_name=pr/>
       <Add mover_name=min/>
        </PROTOCOLS>
</dock_design>
```

e. The protocols section pulls movers we have defined and runs them in order of how they were specified. For this XML script, this mimics a low resolution and high resolution docking moves.

4. Flags file – The flags file for running rosetta scripts is as follows.

```
-docking
        -dock_pert 8 5
        -spin 1
        -randomize1
        -docking_centroid_outer_cycles 50
        -docking_centroid_inner_cycles 500
-docking:dock_mcm_trans_magnitude .1
-docking:dock_mcm_rot_magnitude 1
-native 2VYR_input.pdb
-nstruct  1
-linmem_ig 10
-overwrite
-packing:repack_only
-out:pdb
-use_input_sc
```

You will notice that we have speficied docking flags outside of the XML file. We can do this by using TASKOFSOPERATIONS, Initializefromcommandline, which allows us to pass flags that have not been built into the scripter. This makes it more robust.

5. We now have all files to run protein-protein docking.
    a. $ROSETTA_BIN/rosetta_scripts.$ROSETTA_SUFFIX –database $ROSETTA_DATABSE -parser:protocol $WORKSHOP_ROOT/tutorials/protein-proteindocking/tutorial_files/docking.xml –s $WORKSHOP_ROOT/tutorials/protein-proteindocking/tutorial_files/input._pdb.pdb > log.log

    b. This will only output 1 structure. Usually we want an nstruct of 10,000 to get a good dataset.
6. Analyzing the data.
    a. python $ROSETTA_SCRIPTS/score_vs_rmsd.py –native=2VYR_input.pdb --CA –table=table.out --term=total input_pdb_0001.pdb
    b. using your text editor, open table.out and check your RMSD and score. If you have many files, this table can easily be plotted in a spreadsheet to get an energy funnel.
7. Optional, for graphical mode, add –parser:view to the flags file, a graphical build of rosetta is needed.
    a. If you have the source code, use this command
        i. python scons.py mode=release extras=graphics bin/rosetta_scripts