Quick-start guide to protein-protein docking using Rosetta 3.1

This is one of a series of tutorials designed to get you started with protein docking using Rosetta 3.1. It was produced to accompany Kaufmann et. al. (2010) Biochemistry, and the latest version can be found at http://meilerlab.org/. The purpose of Rosetta's docking mode is to predict the native binding mode between two macromolecules. Note that there is a separate ligand-binding mode for docking small molecules to proteins. The algorithm is described in Gray et. al. J Mol Biol 331, 281-299. It is assumed that you have installed the Rosetta suite from rosettacommons.org, and that you are comfortable working in Linux. You will also want to become familiar with the documentation that can be found in the /manual/ and /demos/ subdirectories, as well as the online Rosetta 3 User Manual, FAQ, and forums at http://www.rosettacommons.org/tiki/.

Steps to execute the docking run

1. You will start by preparing an input PDBfile containing the two proteins you wish to dock. There is a sample file separated.pdb in the /protein_docking/ directory included with this tutorial. It was derived from 1KXQ.pdb (q.v.), a complex of amylase with a camel antibody. For this exercise, the file was simplified to contain only one copy of each chain (labeled A and B) separated by a TER record, with no alternate conformations or missing residues. To simulate a naive (global) docking procedure, the antibody domain was manually moved 20 Angstroms away from the amylase. Docking partner 1 (chain A) is the larger of the two proteins, to improve efficiency.
2. Create a flags file to specify the parameters Rosetta will use during this run, including the location and format of input and output files, and options that specify details of the algorithm. There is a sample flags file in the /protein_docking/ directory included with this tutorial. It is worth reading the file in a text editor becuase it is well-commented and contains pointers to more documentation. The parameters of the run can be changed by editing this file. This flags file has flags appropriate for an initial docking attempt with an antibody; it contains the option -docking::randomize1, which allows full rotation of the amylase, while keeping the antibody domain oriented such that the CDRs face toward the amylase molecule. It also lacks the -out::fullatom flag, resulting in a faster, low-resolution mode in which all sidechains are represented using a centroid approximation.
3. Conduct the folding run:

```
docking_protocol.linuxgccrelease @flags > docking.log
```

Analysis and notes

1. Examine the output files generated during this docking run: the logfile, the output pdb structure files, and the scorefile (docking_output.sc). These are plain text files, so you can read them with a pager or text editor. Explanations of the file formats can be found in the manual. The output structures can be compared using the Rosetta energy score, where the lowest energy indicates the best structure. Per-residue scores can be found at the end of each output structure (or decoy) in the pdbfiles, and total scores are summarized in the scorefile.
2. Load the decoys into your favorite molecular visualization program (Chimera, PyMOL, etc) and compare them visually. You should find the antibody docked all over the amylase protein. By increasing the number of structures (-out::nstruct) to tens or hundreds of thousands, you should be able to identify a preferred binding mode that provides the lowest energies or largest clusters. But where Rosetta truly

shines is in the high-resolution binding mode:

High-resolution mode

1. As the starting structure for a high-resolution run, you can use the output of a low-resolution run, or any complex in which the partners are already near the predicted binding conformation. Rosetta's docking options give you wide control over the amount of perturbation used while sampling the docking pose. For example, with the -dock_pert flag one can specify the angle in degrees the second partner is rotated, and the distance in angstroms it is shifted along the interface, before bringing the partners back together.
2. Conduct a second folding run, and examine the flags2 file to compare the options available for high-resolution mode:
   `docking_protocol.linuxgccrelease @flags2 > docking2.log`
3. Repeat the high-resolution docking run a few times, with different options uncommented in the flags file to get a feel for which options will be best suited to your particular application. For an exhaustive list of all options that the docking application will accept, type **`docking_protocol.linuxgccrelease -help`**