Guide to using PyRosetta and Robetta (Ubuntu and OS X):

This guide will run through how to install and use PyRosetta as well as use Robetta with the end goal of predicting a protein's structure from its fasta sequence and two fragments.

PyRosetta:

1. The first thing that must be done is to install PyRosetta onto the computer you plan on using. This can be done by following the instructions on the download page: http://www.pyrosetta.org/dow
2. N.B. It is recommended to install PyRosetta using Python 3.5, 3.6, or 3.7, as the needed script is written with this syntax regardless of which version you install.
3. (if you do try to install using Python 2.7, open the script "setup.py" in the "setup" folder and change "FileNotFoundError" on line 40 to say "IOError" as the previous syntax is not compatible)
4. Once PyRosetta is correctly intalled, go into the "demo" folder and copy the script "D060_Folding.py" into where your working directory will be.
5. In the same working directory, create the folder ".test.output", as line 116 of the script expects the folder to be ready for the output to go into. If you would like to re-name the outputs folder you need to rename the folder you create and the name of the folder in line 116.

This should be everything you need to do with regards to installing PyRosetta and getting your computer ready to run the code. PyRosetta either accepts a FASTA or a PDB file as an input, so long as two fragment files corresponding to the protein are provided.

The next part of this guide will explain how to create protein fragments for a protein using the online tool Robetta, and is recommended if you want to only fold a small number of proteins.

Robetta:

1. Make an account for Robetta at http://www.robetta.org/register.jsp
2. Get the fasta sequence you want ready to upload.
3. Click on the "Submit" link underneath the "Fragment Libraries" heading.
4. Fill in the required fields, for "Target Name:" put in a reference to find it easily, e.g. the protein name (the optional fields can be left blank) and click submit.
5. Click on the "Queue" option underneath the "Fragment Libraries" heading and your job should be visible as "active". (The time taken to complete it will depend on how many jobs are queued before yours and the length of the proteins, but will be between 20-120mins.
6. Once your job is completed, click on the "completed" link to be taken to the results page.
7. The top two files are the protein fragments of the fasta sequence and regardless of the input protein will be called: aat000_03_05.200_v1_3, and aat000_09_05.200_v1_3 with the highlighted numbers indicating the respective fragment lengths, 3 and 9.
8. Save the two fragments in the ".test.output" folder in your working directory (click each link and press ctrl+s, select the appropriate folder, rename the file of you want, and press save).

9. Also make sure that the FASTA file is in the .test.output folder.

Now that you have the protein fragments, you are ready to run the folding script.

Folding the protein:

These instructions are for running the script from your computer's terminal which can be found by typing "terminal" in your desktop search menu.

1. Open terminal and change working directory to the location of the D060_Folding.py script. This can be done in two ways:

a) If using Ubuntu, locate the folder in your file navigator, right click it, and select "open in terminal".

b) Using the command "cd path/to/working/directory" to get to the folder (Linux Ubuntu or MacOS).

To verify you are in the correct location, type in "ls" to the terminal to see all the files in the current working directory, you should see the ".test.output" folder and the "D060_Folding.py" script.

2. Now, to run the script, type the following into your terminal:

python D060_Folding.py --fasta_filename <name of fasta file> --long_frag_filename <name of protein fragment of length 9> --long_frag_length 9 --short_frag_filename <name of protein fragment of length 3> --short_frag_length 3 –jobs 100 --job_output <job_output_name> --kT 1.0 --long_inserts 1 --short_inserts 3 --cycles 200

Example of what you might type into terminal:

python D060_Folding.py --fasta_filename 2okm.fasta.txt --long_frag_filename aat000_09_05.200_v1_3 --long_frag_length 9 --short_frag_filename aat000_03_05.200_v1_3 --short_frag_length 3 –jobs 1000 --kT 1.0 --long_inserts 1 --short_inserts 3 --cycles 200

If you don't put an option in, the default will be used. Defaults can be found in D060_Folding.py lines 400:446.

All the highlighted parts are options that you can change (you should at least change the names for the input files). Information on the options jobs, kT, and cycles is in the D060_Folding.py script beginning line 557:573.

Once the script has finished running, it will place the predicted strucures in the ".test.output" folder and the terminal will show the filenames with a corresponding score. The lower the score, the better the folding prediction.