3D Data Processing in Structural Biology
Assignment 1
(due April 21st, 2020)

Topics: Molecular Visualization with PyMOL (you may also use Chimera or any other viewer), structure analysis basics, structural alignment with BioPython

The goal of this exercise is to learn how to find structures for proteins of your interest in the protein data bank (PDB), visualize them, and perform basic analyses. You will need to get yourself acquainted with some of these tools**.** You may also choose to consult with available online resources.

We provide some optional references for your convenience -

PyMOL command reference:
https://pymol.org/pymol-command-ref.html

Additional PyMOL resources and tutorials:
https://pymolwiki.org/index.php/Main_Page

PyMOL alignment commands:
https://pymolwiki.org/index.php/Align
https://pymolwiki.org/index.php/Super

BioPython:
https://biopython.org/wiki/Download
https://biopython.org/wiki/The_Biopython_Structural_Bioinformatics_FAQ
http://biopython.org/DIST/docs/tutorial/Tutorial.html
https://warwick.ac.uk/fac/sci/moac/people/students/peter_cock/python/protein_superposition/

We will investigate the COVID-19 virus (SARS-CoV-2) protein complexes. There are currently 3 solved structures for a protein complex between nsp10 and nsp16 (nsp = non-structural protein, not part of the virus shell) with PDB codes: 6w4h, 6w61, 6w75.

1. Which experimental method was used to solve these structures? What can you say about their quality?
2. Open PDB 6w4h in the molecular viewer. The two proteins (nsp10 and nsp16) appear as 2 different chains. Read the PDB header to determine the chain ID for each protein and color nsp10 in red and nsp16 in blue. Save the image in cartoon representation and insert into your exercise.
3. How many alpha-helices are found in each protein? You can color by secondary structure and count or look at the secondary structure annotation on the PDB page under the ‘sequence’ tab

Next, we examine the complex between nsp10 and nsp14. While there is no solved structure yet, there is a model available here:

https://swissmodel.expasy.org/interactive/1zTGNp/models/01

4. Which PDB structure was used as a template for this model? A template is homologous sequence with solved structure. What is the sequence identity between the template and COVID-19 sequence?
5. Download the model, color nsp10 in red and nsp14 in green. Align it to the nsp10-nsp16 complex. Save the image and insert in your exercise.
6. Do you think nso14 and nsp16 can bind nsp10 simultaneously?

Now let's compare to SARS-CoV (2002 coronavirus) and MERS-CoV (2012) nsp10-nsp16 complexes.

7. Find their structures in PDB. You can use text-based search, in case of several structures, prefer the one with higher quality. Align them to the SARS-CoV-2 nsp10-nsp16 complex and save the image of the alignment (using visualization type/tool of your choice).
8. Assessing structural similarity:
    a. Qualitative assessment -
       Describe 1-2 similarities and 1-2 dissimilarities between the various structures. Include 1-2 illustrative images, zooming-in on these similarities and dissimilarities.
       Tip: Be creative in using visual means to make your point. For example, you use different colors or labels, or use different cartoon representations for different amino acids.
    b. Quantitative assessment –
       Root-mean square deviation (RMSD) is a common measure of protein dissimilarity. You can read about it in Wikipedia or search online elsewhere. What is the equation for the RMSD between two sets of corresponding atoms?
    c. What is the RMSD between the structures? Include units.
       Tip: You may use PyMOL align/super commands, or any other available tool to compute the RMSD between structures.
    d. RMSD uses a single number to quantify the dissimilarity between two protein structures with many atoms. Suggest 2-3 possible disadvantages of RMSD.
       Tips: There is not one correct answer so you can be creative. Consider whether you can you get a low RMSD for structures that are dissimilar in some way? Can you get a high RMSD for structures that are similar in some way?
    e. **Bonus:**
       Suggest an alternative measure for RMSD. Include a precise equation. Be creative and discuss one possible advantage and one possible disadvantage of your new measure.

BioPython:

9. Go over the BioPython tutorial, chapter 11, in particular regarding PDB files I/O, structural superimposition, and downloading from the PDB (you may need to go through chapters 1-2 for general context):
   http://biopython.org/DIST/docs/tutorial/Tutorial.html

a. Write a Python script that takes two PDB identifiers and chain identifiers as input. The script will retrieve the appropriate files from the PDB, and align the chains from each file by their set of C-alpha (“CA”) atoms (filtering out residues that lack CA atoms). The output will include the RMSD between the C-alpha atoms (CA) of the two structures, and the aligned PDB files in mmCIF format. It is assumed the two chains have the same number of CA atoms, in the same order. The interface should be:

   ```
   $python my_align.py <pddid1> <chain1> <pdbid2> <chain2>
   ```

   Send us a documented script file. You may optionally include a README file. Make sure the script works on the following example:

   ```
   $python my_align.py 6w4h A 6w61 A
   ```

   The output files in this example are 6w4h.cif and 6w61.cif. Don’t forget to output the RMSD.

   Tip 1: This is not a basic programming course – the script will not go through strict automatic testing, but we do expect it to work under reasonable conditions.

   Tip 2: You may feel free to use any online resources and consult with peers, but **we trust you to write the code yourself from scratch (honor system)**.

   Tip3: Load the output files from your script to make sure the alignment works correctly. Verify that the RMSD is as low as expected.

b. **Bonus 9.1:**

   Add an option to use your alternative measure for RMSD from 8e, or a different measure. Document this measure in the script file, and/or in a README file. The alternative measure will be applied using an optional 5th command line argument, e.g.

   ```
   $python my_align.py 6w4h A 6w61 A True
   ```

   Document your script file to include the words
   “*# BONUS 9.1 IMPLEMENTED*”.

c. **Bonus 9.2:**

   BioPython’s superimpose() command assumes an equal number of atoms. Have your script (a) apply to chains of any length, by first applying sequence alignment between the two PDB files prior to their structural alignment.

   Make sure the script works on the following example:

   ```
   $python my_align.py 6w4h A 6w75 A
   ```

   Document your script file to include the words
   “*# BONUS 9.2 IMPLEMENTED*”.

   Tip: You may wish to consult online resources, such as the relevant sections on sequence alignment in the BioPython tutorial. This short tutorial may also come handy, at least as inspiration:
   https://warwick.ac.uk/fac/sci/moac/people/students/peter_cock/python/protein_superposition/

Good Luck!