**Training plan: scanning for proline-compatible positions in the protein structure**

Hongjun Bai @ VGS, MHRP

July 13, 2020

**[Prerequisite knowledge/skills]**

**Basics of amino acid, protein and protein structure**

Key concepts: amino acid, side chain, classification of amino acids, peptide bond, **alpha-helix, parallel/antiparallel beta sheet, primary/secondary/tertiary/quaternary structure**, non-covalent interactions (hydrophobic interactions, hydrogen bonds, salt-bridge, van der Waals interaction), disulfide bond

* <https://www.khanacademy.org/science/biology/macromolecules#proteins-and-amino-acids>
* <https://www.khanacademy.org/test-prep/mcat/biomolecules#amino-acids-and-proteins1>
* dihedral angles in the backbone: Ψ and Φ
* Accessible surface area (ASA) of residues

**Necessary computational skills**

* Operation system: Mac OS or Linux
  + Ubuntu/Debian
  + The file tree
* Command line environment – useful commands (not a complete list)
  + echo, ls, cp, mv, mkdir, find, history, pwd, ln, du, df, top, kill, sudo, su, ps, chmod, less, man, which
  + wc, head, tail, cat, grep, diff, sort, uniq, split, sed, cut
  + git, pacman/apt-get, nano/vim, awk
* Proficiency in python programming language
  + python 3.8: <https://docs.python.org/3.8/tutorial/>
* PyCharm, Git, GitHub
  + https://clt.champlain.edu/knowledgebase/configuring-git-with-pycharm/
  + Install git program
    - Necessary commands - diff, clone, revert, commit, push, pull
  + Create a GitHub account, set up a repository, integrate the Git, GitHub, PyCharm and practice the usage
    - Same time while learning python unittest, implement warm up task 1 and 2.
* Test driven development
  + Python unittest
    - <https://www.digitalocean.com/community/tutorials/how-to-use-unittest-to-write-a-test-case-for-a-function-in-python>
  + Warm up task 1 and 2 as test cases
* Necessary python models
  + Biopython PDB
    - <http://biopython.org/DIST/docs/tutorial/Tutorial.html#sec182>
    - <https://biopython.org/wiki/The_Biopython_Structural_Bioinformatics_FAQ>
  + numpy
  + Matplotlib
    - <https://matplotlib.org/>
  + Pandas

**[Additional bioinformatics tools and online resources]**

* pymol
  + windows install: <https://pymolwiki.org/index.php/Windows_Install>
  + Tutorial part 1: <https://www.youtube.com/watch?v=ai7p9Neguks>
  + Tutorial part 2: <https://www.youtube.com/watch?v=uxa-9UYnIAw>
* RCSB PDB
  + URL: <https://www.rcsb.org/>
  + PDB format: <https://www.cgl.ucsf.edu/chimera/docs/UsersGuide/tutorials/pdbintro.html>
* DSSP – secondary structure assignment
  + Maybe too much work to get it installed on windows, use the webserver instead.
  + <https://swift.cmbi.umcn.nl/gv/dssp/DSSP_3.html>
  + <https://github.com/PDB-REDO/dssp>
  + Webserver: <https://www3.cmbi.umcn.nl/xssp/>
* Aliview, the alignment viewer
  + URL: <https://ormbunkar.se/aliview/>

**[The project]**

**scanning for proline-compatible positions in the protein structure**

**Timeline** (to be considered)

2-3 weeks: learn the basics, lay out the problem

3-4 weeks: working on the problem

1-2 weeks: wrap up / poster presentation

**TASKS**

Warm up tasks (to be updated):

1. Function to calculate distance between two points/atoms in 3D space
2. Function to calculate dihedral angle for four given points/atoms in 3D space
3. Clean up non-amino acid residues in a PDB structure
4. Read and parse PDB, get the xyz of a given atom in each residue
5. Calculate Ψ and Φ angles for a given site in the structure
6. Read and parse DSSP output
   * Extract accessible surface area (ACC/ASA), phi, psi and secondary structural assignment a specific site

Tasks

* Superimpose two residues by their backbone atoms (CA, C, N and O)
  + Bio.PDB.Superimposer
  + Test function: compare with superimposition result from pymol (or a man-made example)
  + RMSD for the evaluation of superimposition
    - <https://en.wikipedia.org/wiki/Root-mean-square_deviation>
* Typical proline conformations in high resolution structures. Extract proline residues from a given structure (7DWY), check
  + Phi, psi angles (two different combinations)
  + Side chain atom positions (two different positions)
  + Save the information
    - Typical structures
    - A plot of proline conformations: 3-axis, x - phi, y - psi, and z - chi; expecting four clusters.
      * chi – dihedra angle formed by ‘N-CA-CB-CG’
* Check backbone compatibility of proline
  + A class to handle proline backbone compatibility
    - Initialization: take the trans proline phi-psi contour map as data
    - A member function:
      * Input: phi-psi pairs
      * Output: the corresponding probability of the given phi-psi pair in the Ramachandran plot
      * Test – choose a phi-psi pair from the phi-psi contour map, check if the function returns correct probability
    - Trans proline phi-psi contour (Ramachandran plot):
      * [rama8000-transpro.data](https://github.com/rlabduke/reference_data/blob/master/Top8000/Top8000_ramachandran_pct_contour_grids/rama8000-transpro.data)
      * data source
        + Publication: <https://pubmed.ncbi.nlm.nih.gov/27018641/>
        + Data: <https://github.com/rlabduke/reference_data>
      * Probability cutoff: p > 0.01
  + A function to return backbone compatible sites of a given structure
    - Input: a structure (Bio.PDB)
    - Output: residues on the backbone compatible sites
    - Test function:
      * Expectation: take a manageable segment from a structure (a 50 AA fragment?), check the phi-psi and corresponding probability in the Ramachandran plot, note down all the sites that with p > 0.01
  + A member function to return typical conformations
* A class to handle proline conformations
  + Initialization: load the typical conformations of proline
    - Organized as a dictionary
      * key - (phi, psi)
      * value - the proline residue
  + A member function:
    - Input: phi-psi pair
    - Output: proline with (phi, psi) pair closest to the given phi-psi pair
    - Test
      * Expectation: given a compatible phi-psi pair, get the proline with most similar backbone conformation returned
        + “most similar” – minimum distance between the (phi, psi) and the (phi, psi) of pro in the stored dictionary
  + NOTE: used in the function to mutate a given site to proline
    - For now, use all prolines from 7DWY.
    - will be finalized later, after a more rigorous check of typical proline conformations.
* A function to mutate a given site to proline
  + Input: a structure, a residue is checked that is backbone compatible with proline
  + Output: the structure with the given residue mutated to proline
  + NOTE:
    - before replacing the selected residue, modify the residue ID of proline that is returned by the class that handling proline conformations to be identical to the selected residue
  + Test function:
    - Select a proline in a given PDB structure, remove side chain atoms and rename it as GLY. Then, include that proline (after a randomly move, e.g. backbone superimpose with an random residue in the PDB structure) in the proline conformation list. Carry out the mutation, see if the proline in the original structure gets restored (identical residue id and side chain atom xyzs).
* Calculate the cost of the proline replacement
  + Count number of neighbor atoms of a given residue
  + ACC
  + SSE from i-3 to i+3
* Compare automatically identified sites to the 6 proline locations of SARS-CoV-2 splike-6P

**Routine meeting:**

Daily (Mon-Thu, 10 am)