Session I. Introduction to protein structure manipulation in python

Objective

To acquire the basic skills to load/save a protein structure in PDB format, identify their components, obtain basic information, and perform geometric measures on the structure

Software and libraries	• Python >= 3.7 • Biopython module (>= 1.72) • Molecular viewer (pymol, chimera) • (optional) Jupyter notebook, nglview)
Code Examples	https://github.com/jlgelpi/Biophysics
Biopython reference	• Biopython tutorial • Bio.PDB tutorial • Biopython reference

Conda installation (recommended)

Requires Anaconda or miniconda installed.

1. Create new environment and activate it
```
conda create -n your_env_name
conda activate your_env_name
```

2. Install Biopython
```
conda install biopython
```

3. Install Notebook and nglview (optional)
```
conda install jupyter nglview
```

Simple Examples

https://github.com/jlgelpi/Biophysics/tree/master/Examples

- ex_cmd_line.py: Simple command line
- ex_distances.py: Search for contacts
- ex_distances_2res.py: Print distances between atoms
- ex_list_res.py: Print atoms and coordinates for ARG residues
- ex_list_res2.py: Print residue atoms of a residue number
- ex_chains.py: Remove a list of chains and save the remaining in a PDB file
- Biopython_Examples.ipynb (Notebook containing above examples)

Exercises

Prepare scripts for exercises below using ***argparse*** to build the appropriate command line.

Output lists should be sorted (by residue o atom number) when appropriate and formatted for an easier read.

Upload a **tar.gz** file with codes (or github link) and **examples of the output** for each exercise. Output should be properly formatted.

1. Determine the list of pairs of residues whose CA atoms are closer than a given distance
   Parameters: PDB file name, distance.
2. Generate a list of all atoms for a given residue number
   Parameters: PDB file name, Residue number (Including Chain if applicable)
3. Determine all possible hydrogen bonds (Polar atoms at less than 3.5 Å).
   Parameters: PDB file name. Optional: cut-off distance (defaults to 3.5)
4. Generate a list of all CA atoms of given residue type with coordinates
   Parameters: PDB file name, residue type.
   Optional: accept residue codes in one- or three-letter formats automatically
5. Generate a list of backbone connectivity (i.e. which residues are linked by ordinary peptide bonds).
   Parameters: PDB file name. Optional: Cut-off distance for peptide bonds (defaults to 2.5)
6. Id 4, but for disulphide bonds.
7. Print distances between all atom pairs of two given residues
   Parameters: PDB file name, Residue 1, Residue 2

Hints

General

Bio.PDB uses a hierarchical set of lists to store the structure

```
Structure
  |- Models
       |- Chains
           |- Residues
               |- Atoms
```

Each level holds a list of elements of the child level (i.e. a Chain is a list of Residues)

Each element has a specific .id element:
Models: Number of model (integer), Chains: chain id (A, B,…), case sensitive, Residues: Tuple where the second element is the residue number, i.e. (‘’,24,’’). Atoms: Atom name

Any element can be accessed directly using the ids: Structure[0][‘B’][24][‘CA’] would correspond to Atom **CA** in residue number **24** of Chain **B** of Model **0**

.get_parent() for all elements gives the corresponding parent element.

.get_residues() , .get_atoms() provide residues or atom lists for all elements

Each element has specific fields and functions with the information regarding that element, check Bio.PDB reference

https://biopython.org/docs/1.75/api/Bio.PDB.Atom.html,
https://biopython.org/docs/1.75/api/Bio.PDB.Residue.html,
https://biopython.org/docs/1.75/api/Bio.PDB.Chain.html

Hints for Exercises

Argparse. Follow ex_cmd_line.py for a simple command line. Adapt the example to the required parameters for the following exercises.

Output format. Feel free to format the output as desired, as long as it contains the required information. You may use residue_id() and atom_id() functions (defined in the example Notebook) to your convenience.

1. Use ex_distances.py as a template. Select CA atoms only for each residue and either 1) iterate over all pair of residues and check distances, or 2) use the NeighborSearch module as in the example

2. Follow ex_list_res.py example

3. Adapt exercise 1 to select pairs of polars atoms (O, N, S) that form a possible Hydrogen bond, using a distance criterium (dist < 3.5 Å)

4. Iterate over all residues and select those matching the required residue type. For residues selected, print the required information.

5. Ordinary peptide bonds are made between atom C of one residue and atom N of the following, Usual distance should be below 2 Å. Following the same approach in exercises 1, or 3, find pairs of C-N atoms from different residues that are closer than 2 Å.

6. Disulphide bons are formed between S atoms of Cys Residues when they are at the appropriate distance (around 1.9 Å). Using the same approach as 1, 3, or 5, find S-S contacts. Allow some more distance to access structure variability.

7. Select the desired residues and follow ex_distances_2res.py example.