Assignment 2: Parsing a text file and simple calculations in Python

In this assignment you will parse a text file describing a protein structure and do some simple calculations on the atomic positions. Obtaining the 3D structure of a protein provides important clues to how the protein functions and interacts with small molecules, DNA and other proteins. It is however a major experimental enterprise. The text file comes from the Protein Data Bank (PDB, www.pdb.org), the standard database for protein structures. The PDB is an open repository for experimentally determined structures of proteins, now with over 100,000 entries. Files with experimental details and the atomic positions can freely be downloaded in the .pdb format.

In this assignment you will write code to parse the file, and retrieve part of the information, particularly the resolution of the experiment and the atomic coordinates. Then you will explore the distribution of hydrophobic residues relative to the core of the protein, compared to the other residues. Finally,your programwill calculate which amino acid residues are closest to the ligand of the protein.

Part I

Find on Toledo the file 5kkk.pdb, this is the structure file in PDB format you will be working with. It contains the structure for myoglobin of the sperm whale. The position of the protein's main ligand, the heme group, was also determined.

Have a look at the file, the coordinates are marked by the ATOM keyword. The so-called alpha carbon, marked as CA, is a central atom for every amino acid along the protein backbone. For simplicity, you will focus only on these atoms, and use them to represent the positions of the amino acids in the structure. Below you find an example of the part of the file that is of interest:

ATOM	1085	N	GLY	A	130	58.858	-11.439	-0.233	1.00	17.21	N
ATOM	1086	**CA**	**GLY**	A	130	**59.959**	**-11.275**	**0.707**	1.00	16.16	C
ATOM	1087	C	GLY	A	130	60.365	-12.589	1.372	1.00	16.45	C
ATOM	1088	O	GLY	A	130	61.556	-12.904	1.460	1.00	16.85	O
ATOM	1089	N	ALA	A	131	59.386	-13.357	1.845	1.00	15.10	N
ATOM	1090	**CA**	**ALA**	A	131	**59.706**	**-14.624**	**2.520**	1.00	16.75	C
ATOM	1091	C	ALA	A	131	60.321	-15.631	1.542	1.00	18.89	C
ATOM	1092	O	ALA	A	131	61.285	-16.319	1.888	1.00	14.71	O
ATOM	1093	CB	ALA	A	131	58.461	-15.215	3.178	1.00	18.88	C

Make a new Python module, and add the function parsePDB. It should accept as a parameter a string representing the path to the PDB file. It should return three objects: 1. a list of the positions of the Ca atoms of the amino acids. 2. The accompanying residues in order. 3. A list with the entries (the lines) for all heterogens, the atoms that are associated with the protein during the crystallography, but that are not an integral part of it.

Write the code to parse the file yourself (don't use some shared package). Take into account the ATOM field information as provided by PDB, see below. In the table, columns are simply the position in the character string for every line; meaning characters 1 to 6 are the first field, so ATOM followed by two spaces 'ATOM '.

Overview: The ATOM records present the atomic coordinates for standard amino acids and nucleotides. They also present the occupancy and temperature factor for each atom.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"ATOM "	
7 - 11	Integer	serial	Atom serial number.
13 - 16	Atom	name	Atom name.
17	Character	altLoc	Alternate location indicator.
18 - 20	Residue name	resName	Residue name.
22	Character	chainID	Chain identifier.
23 - 26	Integer	resSeq	Residue sequence number.
27	AChar	iCode	Code for insertion of residues.
31 - 38	Real(8.3)	x	Orthogonal coordinates for X in Angstroms.
39 - 46	Real(8.3)	y	Orthogonal coordinates for Y in Angstroms.
47 - 54	Real(8.3)	z	Orthogonal coordinates for Z in Angstroms.
55 - 60	Real(6.2)	occupancy	Occupancy.
61 - 66	Real(6.2)	tempFactor	Temperature factor.
77 - 78	LString(2)	element	Element symbol, right-justified.
79 - 80	LString(2)	charge	Charge on the atom.

Non-polymer or other "non-standard" chemical coordinates, such as water molecules or atoms presented in HET groups use the HETATM record type.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"HETATM"	
7 - 11	Integer	serial	Atom serial number.
13 - 16	Atom	name	Atom name.
17	Character	altLoc	Alternate location indicator.
18 - 20	Residue name	resName	Residue name.
22	Character	chainID	Chain identifier.
23 - 26	Integer	resSeq	Residue sequence number.
27	AChar	iCode	Code for insertion of residues.
31 - 38	Real(8.3)	x	Orthogonal coordinates for X.
39 - 46	Real(8.3)	y	Orthogonal coordinates for Y.
47 - 54	Real(8.3)	z	Orthogonal coordinates for Z.
55 - 60	Real(6.2)	occupancy	Occupancy.
61 - 66	Real(6.2)	tempFactor	Temperature factor.
77 - 78	LString(2)	element	Element symbol; right-justified.
79 - 80	LString(2)	charge	Charge on the atom.

Part II

Make a new function distance2Center in the module that takes a list of coordinates as a variable, and returns for every coordinate the distance to the center of the protein. Calculate the center of the protein by taking the centroid of the Ca coordinates (the average position over the positions r: $\bar{r} = \frac{1}{n}\sum_{i=1}^{n} r_i$. Then for all coordinates calculate the distance to this center: $\|\bar{r} - r_i\|$. [1]

Write code in your script to output the result of this function to a file. Your information should include: for every residue the distance, the residue information (three letter code as parsed in Part I), and a binary marker indicating if the residue is hydrophopic or not. The hydrophobic amino acids are: ALA, CYS, PHE, ILE, LEU, MET, PRO, VAL and TRP.

Now, using matplotlib, plot the distances for both the hydrophopic amino acids and the remainder. You can make a boxplot, a line plot, even comparing the two distributions. Comment on your result.

Part III

Now make a new function in your module dubbed searchHeterogen. It accepts the list of HETATM from Part I and an Atom name. It returns a list of position coordinates. Use this function to retrieve the position for atom "FE", the iron molecule in the hemoglobin. Now write some code to print the five amino acids to screen, represented by their Ca location, that are nearest to this atom (include index or the residue, distance and residue type). Does one of these residues indeed interact with the heme group? (Hint: Look at the PDB entry online!)

[1] HINT: Make a function to calculate the distance between two coordinates. To take the square root you should import the math or NumPy library: import math as m;m.sqrt(2)

Report:

In your assignment report give the answers to the question, and show the graph for Part II. Also provide the script and your module. Please bundle the files in a single .zip or other archive before uploading.