Université Paris Cité

SCHADTLER-LAW
M2 BI

2022-2023

Determination of the accessible solvent area (ASA) of proteins from protein coordinates through python programming.

Introduction

In protein research, identifying surface residues is a key factor in understanding it's stability, interactions and potential druggability. However, a surface residue isn't necessarily accessible by surrounding molecules. Therefore, the accessible solvent area (ASA) of a protein is often more relevant than its total surface. The ASA was first been defined by Lee and Richards (1971) [1].This parameter is also used in biomolecules' energy characterization. The ASA of a protein is calculated by adding the solvent radius in the calculation of the atom's surface. Here we intend to build an efficient program able to calculate the ASA, the relative ASA (rASA) of the protein, rASA per residue and the accessibility percentage from the protein's coordinates.

Materials and methods

Materials

This program is written in python 3.9 and executable in a Linux environment. The script, environment's description file and user manual are accessible in a GitHub repository: (https://github.com/JaneSLW/M2_projet_court).

Several external python packages are required. BioPython (version 1.79) package was used for the Protein Data Bank (PDB) query and parsing of the ".pdb" file [2]. Numpy (version 1.23.1) package helped with matrix handling and overall rapidity of the processing [3]. Scipy (version 1.9.1) was used for the distance matrix calculation.

The program requires either the PDB identifier or a .pdb file. In both cases the program uses the 3D protein coordinates written in .pdb file format. The proteins taken as examples are:

- Collagen ("IBKV") [4]
- Alpha beta tubuline dimer ("1TUB") [5]
- Human Keratin 1/10-1B ("6EC0") [6]
- Human serum albumin ("1AO6") [7]
- Human E-Cadherin ("4ZT1") [8]
- Antimicrobial peptide L-K6 ("6A5J") [9]

The proteins selected vary greatly in size on purpose.

Method

The program is based on the creation of a sphere around each atom. The sphere is composed of 92 points evenly distributed. As mentioned by Shrake and Rupley (1973), increasing the number points of the sphere from 92 to 400 won't significantly affect the ASA [10]. The sphere's radius equals the sum of Van der Waals radius of the atom and the solvent radius. Therefore, it resembles a probe (in this case: a water molecule) rolling over the protein.

After parsing the .pdb file, the program keeps in memory several information: the atoms' sequence, the residues' sequence, the atom's parent residues and the atoms' coordinates (Figure 1). From the coordinates it generates the distance matrix which will inform on the neighboring atoms. Then for each point of the sphere and for each atom it calculates the distance between the point and the center of the neighboring atom. If this distance is inferior to the sum of the radius of the neighboring atom and the water radius, then this point is considered buried. In order to optimize the program, when a point is too close to neighboring atoms, the loop stops and the point is directly considered buried. Afterwards, with all the "surface points", the accessible solvent area can be calculated by multiplying the sphere's surface with the number of surface points divided by the total of points (92).

The rASA of each residue was calculated with the residues' maximum possible solvent accessible surface area (MaxASA) determined by Tien and al. (2013) [11]. The ASA value of each residue was normalized by its MaxASA.

The solvent accessibility percentage was calculated, as described in Shrake and Rupley (1973), by dividing the total ASA with the total area of solvated spheres.

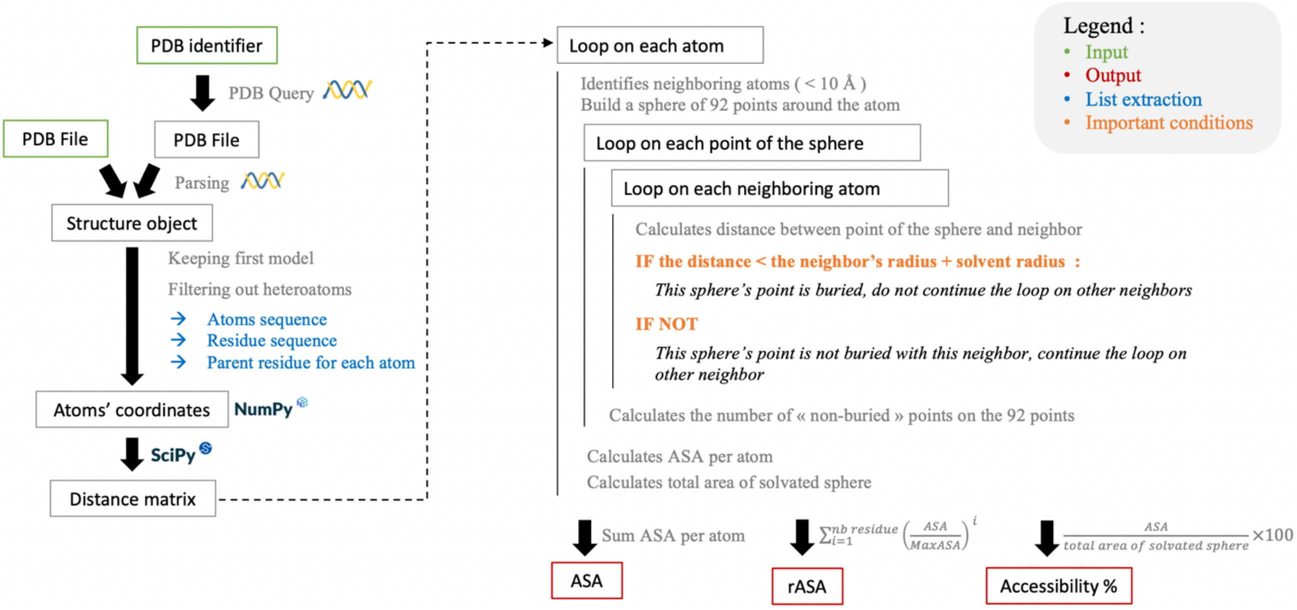
PDB identifier
PDB Query
PDB File
PDB File
Parsing
Structure object
Keeping first model
Filtering out heteroatoms
→ Atoms sequence
→ Residue sequence
→ Parent residue for each atom
Atoms' coordinates
NumPy
SciPy
Distance matrix
Loop on each atom
Identifies neighboring atoms (< 10 Å)
Build a sphere of 92 points around the atom
Loop on each point of the sphere
Loop on each neighboring atom
Calculates distance between point of the sphere and neighbor
IF the distance < the neighbor's radius + solvent radius :
This sphere's point is buried, do not continue the loop on other neighbors
IF NOT
This sphere's point is not buried with this neighbor, continue the loop on other neighbor
Calculates the number of « non-buried » points on the 92 points
Calculates ASA per atom
Calculates total area of solvated sphere
Sum ASA per atom
$\sum_{i=1}^{nb\ residue}\left(\frac{ASA}{MaxASA}\right)^{i}$
$\frac{ASA}{total\ area\ of\ solvated\ sphere}\times 100$
ASA
rASA
Accessibility %
Legend :
• Input
• Output
• List extraction
• Important conditions

Figure 1. Non-exhaustive figure of the different steps of the program.

Results

The following results have been generated with the bash command:

$ python src/main.py -n 6A5J

For the selected proteins, their ASA results have been compared to their results obtained with DSSP (Define Secondary Structure of Proteins, Kabsch and Sander 1983) [12]. The results are shown in Table 1 and Figure 2. An error percentage of the developed program has been determined and is shown in Table 1.

Protein	Size (atoms)	ASA in Å (Program)	ASA in Å (DSSP)	Error Percentage
6A5J	109	1587,58	1556	2,03%
6RQS	195	2443,7	2462	0,74%
3I40	395	3295,85	3437	4,11%
1BKV	563	5820,61	5808	0,22%
6EC0	1731	15880,98	16276	2,43%
4ZT1	3180	21447,71	22175	3,28%
1TUB	6849	31266,76	33812	7,53%
1AO6	9198	54126,95	56907	4,89%

Table 1. ASA results obtained with DSSP and the program developed.

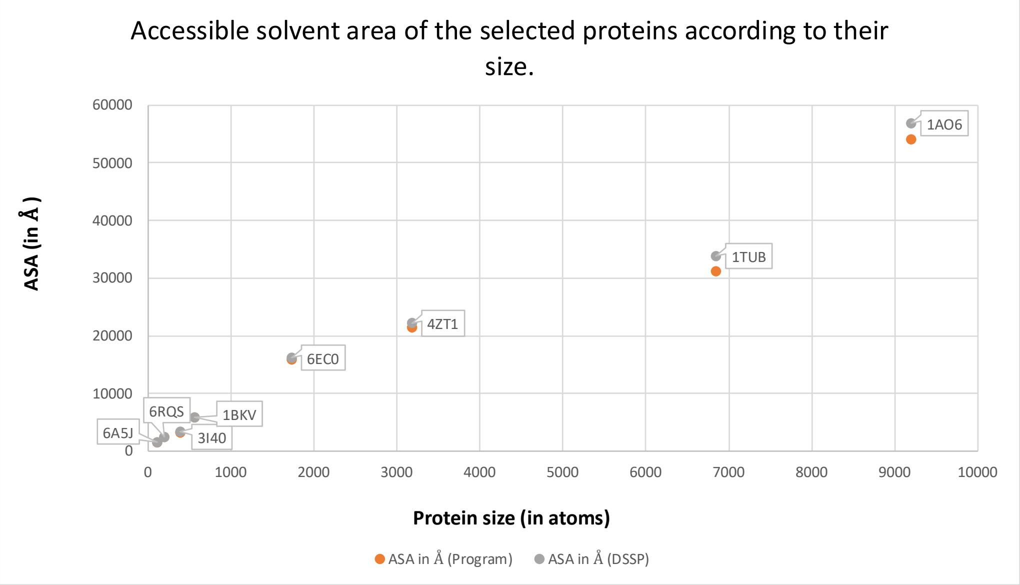

Figure 2. Dot plot of the ASA values obtained with DSSP and the teste program according to the protein size.

The error percentage of the developed program varies from 0.74% to 7.53% between the proteins. The two largest proteins have the higher error percentage. The difference between the results of the two programs are more noticeable for larger proteins.

Discussion

The program is slightly slower than DSSP for large proteins (1TUB and 1AO6). However, for smaller proteins the program, like DSSP, provides the results instantly. The larger the protein is, the less accurate the program is. This could be due to a higher precision of DSSP which uses a 320 points sphere [13]. The number of points in the sphere could be increased in the developed program but only at the expense of its rapidity of execution. DSSP is written in C++, hence it compiles a lot faster.
In order to overcome its inaccuracy, the program should integrate a function to execute time-consuming calculations simultaneously in multiple processors and the number of points that defines the sphere should be increased.

Bibliography

[1] B. Lee et F. M. Richards, « The interpretation of protein structures: estimation of static accessibility », *J. Mol. Biol.*, vol. 55, n° 3, p. 379‑400, févr. 1971, doi: 10.1016/0022-2836(71)90324-x.

[2] « Biopython · Biopython ». https://biopython.org/ (consulté le 14 septembre 2022).

[3] « NumPy ». https://numpy.org/ (consulté le 14 septembre 2022).

[4] R. P. D. Bank, « RCSB PDB - 1BKV: COLLAGEN ». https://www.rcsb.org/structure/1BKV (consulté le 14 septembre 2022).

[5] R. P. D. Bank, « RCSB PDB - 1TUB: TUBULIN ALPHA-BETA DIMER, ELECTRON DIFFRACTION ». https://www.rcsb.org/structure/1TUB (consulté le 14 septembre 2022).

[6] R. P. D. Bank, « RCSB PDB - 6EC0: Crystal structure of the wild-type heterocomplex between coil 1B domains of human intermediate filament proteins keratin 1 (KRT1) and keratin 10 (KRT10) ». https://www.rcsb.org/structure/6ec0 (consulté le 14 septembre 2022).

[7] R. P. D. Bank, « RCSB PDB - 1AO6: CRYSTAL STRUCTURE OF HUMAN SERUM ALBUMIN ». https://www.rcsb.org/structure/1ao6 (consulté le 14 septembre 2022).

[8] R. P. D. Bank, « RCSB PDB - 4ZT1: Crystal structure of human E-Cadherin (residues 3-213) in x-dimer conformation ». https://www.rcsb.org/structure/4zt1 (consulté le 14 septembre 2022).

[9] R. P. D. Bank, « RCSB PDB - 6A5J: solution NMR Structure of small peptide ». https://www.rcsb.org/structure/6A5J (consulté le 14 septembre 2022).

[10] A. Shrake et J. A. Rupley, « Environment and exposure to solvent of protein atoms. Lysozyme and insulin », *J. Mol. Biol.*, vol. 79, nº 2, p. 351-371, sept. 1973, doi: 10.1016/0022-2836(73)90011-9.

[11] M. Z. Tien, A. G. Meyer, D. K. Sydykova, S. J. Spielman, et C. O. Wilke, « Maximum Allowed Solvent Accessibilites of Residues in Proteins », *PLoS ONE*, vol. 8, nº 11, p. e80635, nov. 2013, doi: 10.1371/journal.pone.0080635.

[12] « DSSP ». https://swift.cmbi.umcn.nl/gv/dssp/ (consulté le 14 septembre 2022).

[13] W. Kabsch et C. Sander, « Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features », *Biopolymers*, vol. 22, nº 12, p. 2577-2637, déc. 1983, doi: 10.1002/bip.360221211.