Conception of a python program to calculate the Solvent-Accessible Surface Area (SASA) of a protein by using its atomic three-dimensional coordinates

Souad YOUJIL ABADI

souad.youjil-abadi@etu.u-paris.fr

Master 1 BI-IPFB : Biologie-Informatique et Ingénierie de Plate-forme en Biologie

2022 - 2023

Projet court :

Tatiana Galochkina tatiana.galochkina@u-paris.fr

Jean-Christophe Gelly jean-christophe.gelly@u-paris.fr

1 INTRODUCTION

The solvent-accessible surface area (SASA) or accessible surface area (ASA) is the surface area of a biomolecule that is accessible to a solvent, usually described in units of Å^2 (a standard unit of measurement in molecular biology) [1], and is a decisive parameter for studying the folding of polypeptide chains. In this project, a computer program is created for calculating the exposure of atoms of a protein to solvent (H2O), by implementing the Shrake and Rupley algorithm (1973)[2]. The computation is based on the atomic three-dimensional coordinates derived from the different methods implemented in the PDB (Protein Data Bank) database to determine the 3D structure of a protein, (e.g. X-ray crystallography, NMR spectroscopy and electron microscopy).
It is also based on assumptions like those of Lee and Richards (1971), who were the first to describe ASA (also called as Lee-Richards molecular surface) and develop a computer program for calculating the exposure of protein atoms to solvent. [3] ASA is now obtained by computer programs such as Naccess or DSSP [4]. The latter has been used as a comparator to our program measurements.

Principle:

Like Lee & Richards (1971) and Shrake & Rupley (1973), the protein is described by a set of solvated van der Waals spheres. The surface of a sphere is represented by a set of 92 test points that are nearly uniformly distributed. Each atom of the protein is considered separately as a central atom that is checked for overlap with all other atoms of the molecule [2] **(Figure 1)**. Like Lee & Richards (1971) and Shrake & Rupley (1973), hydrogen atoms are not explicitly considered, since they are incorporated into the van der Waals radii for groups. [2] 92 points represent the surface of the solvated sphere with sticient accuracy for this type of calculation.[2]

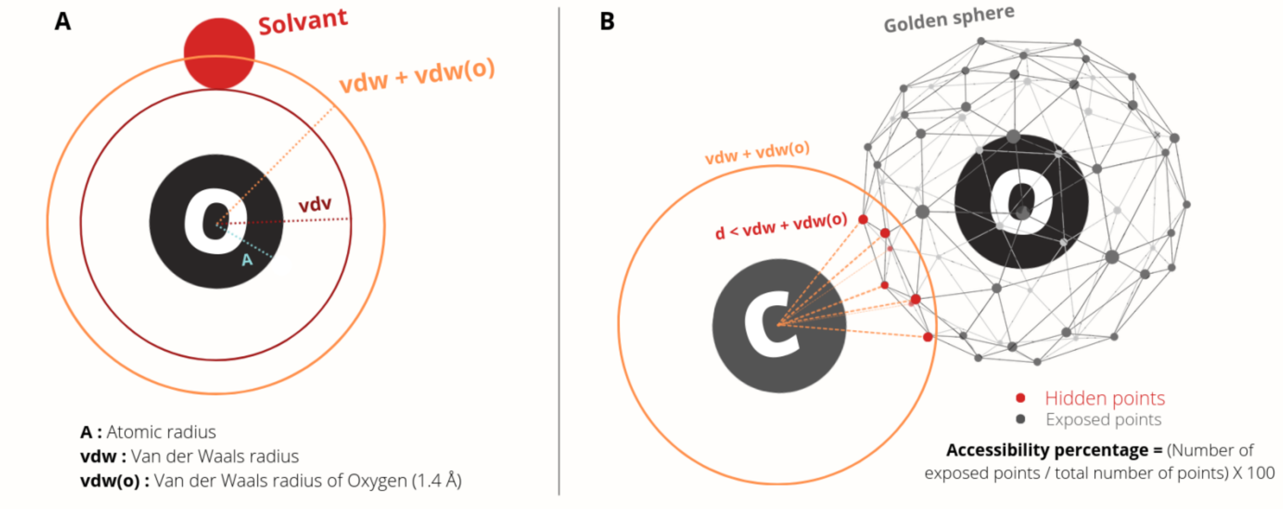

A
Solvant
vdw + vdw(o)
O
vdv
A
A : Atomic radius
vdw : Van der Waals radius
vdw(o) : Van der Waals radius of Oxygen (1.4 Å)
B
Golden sphere
vdw + vdw(o)
d < vdw + vdw(o)
O
C
Hidden points
Exposed points
Accessibility percentage = (Number of exposed points / total number of points) X 100

Figure 1: Description of the Shrake & Rupley algorithm. A: Solvated van der Waals sphere. B: Interaction between two solvated van der Waals spheres. These spheres have as a radius the VDW radius of the considered atom + VDW radius of oxygen (solvent).

2 MATERIAL AND METHODS

2.1 MATERIAL

Data:

Three .PDB files (from which the 3D atomic coordinates (x, y, z) have been extracted) have been downloaded from the PDB database and used as examples to execute the program. These are 1bzv.pdb (46 residues: 366 atoms), 1rex.pdb (130 residues, 1135 atoms), 1si4.pdb (574 residues, 4379 atoms), corresponding respectively to synthetic-made insulin analogue, human lysozyme and human hemoglobin.

Programming language:

The programming language used to code the presented program is Python3.

Conda and Python external libraries:

With the conda package manager, provided with Anaconda, additional Python modules have been installed. The following external Python libraries have been used:

NumPy (for vector and matrix manipulations, linear algebra), **Pandas** (for DataFrames, data analysis), **Matplotlib** (for graphical representations), **Argparse** (for command-line arguments, allowing user input values to be parsed and utilized), **Tqdm** (for creating progress meters or progress bars), **Tabulate** (to display table data easily) and Joblib (for parallel computing).

Github:

All the codes implemented in this software have been deposited on the github repository whose link is: https://github.com/souadyoujilabadi/SASA_project

2.2 METHODS

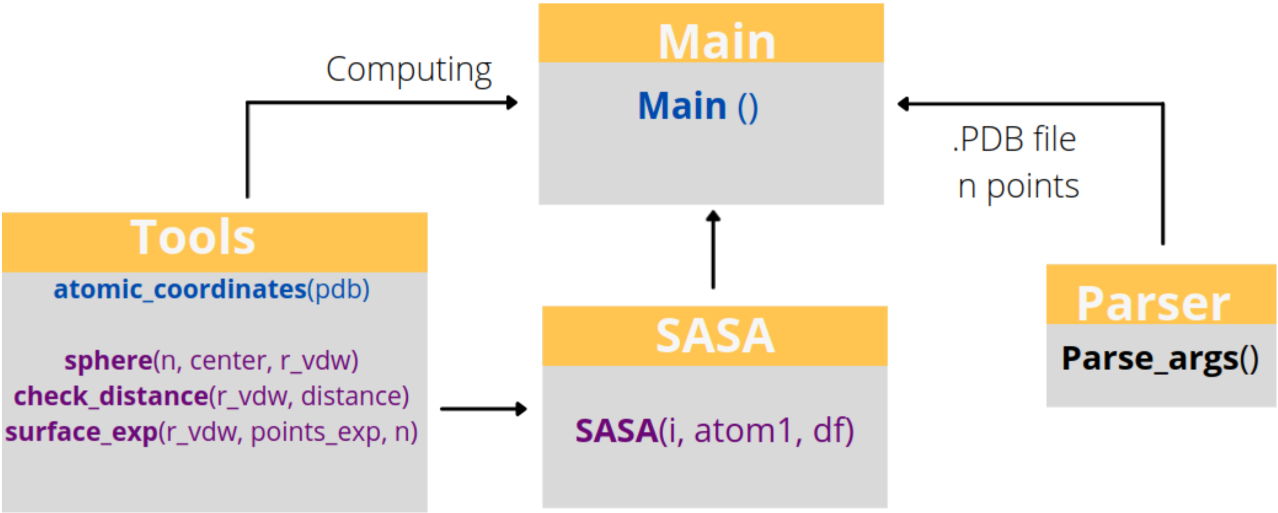


Figure 2: Description of the program. The library called tools is used by the functions SASA() and main(). Parse_args() allows user input values (here, the pdb file and the number of points) to be parsed and utilized by the program.

Program principal function - Main():

Initially, the arguments entered by the user (PDB file and number of points to be evenly distributed on the spheres, with the options -pdb and -n respectively) are taken by the program, which then reads and extracts the atomic coordinates of the protein atoms using the function **atomic_coordinates**. This latter returns a DataFrame containing the name and 3D coordinates associated with each atom, as well as the residue name and number.

In a second step, the function SASA(), which calculates the ASA and the total surface area of each atom, is used by the program in a parallelized way (i.e. it's runned in parallel; here the DataFrame was divided by 4 and the SASA() function was runned on each part in parallel).

Finally, for each residue, the ASAs obtained for each atom are merged/added, and the protein's accessible surface area and percentage of accessibility are then calculated.

SASA():

The functions available in the module tools are used by the function SASA() as follow:

A sphere is generated for an atom using the function **sphere()** (which evenly distributes the number of points entered by the user on the sphere using the Golden spiral algorithm).

The distance between this atom and the other protein atoms is then calculated and only the atoms located in the neighborhood (using a threshold of 10Å) are selected.

The distance between the points of the considered atom and the nearest selected atoms is calculated and checked using the function **check_distance()** (which checks if this distance is < to the considered atom Van der Waals radius + 1.4Å (the VDW radius of oxygen (solvent). If so, the points are hidden, i.e. non exposed to the solvent). The exposed surface (ASA) is then calculated by the function **surface_exp()**, which converts the number of the exposed points of the considered atom to Å^2 by multiplying the exposition ratio (points exposed/total points) to the total surface area of the sphere (4 * π * (sphere radius = VDW radius of the atom + 1.4)^2.

3 RESULTS AND DISCUSSION

3.1 Measurements

The accessible surface area obtained with the programs SASA and DSSP has been plotted for each residue of the proteins 1bzv (Figure 3) and 1rex (Figure 4).

These two scatterplots show a strong correlation between the measurement obtained with the developed program (SASA) and the program used as a comparator (DSSP). Indeed, the estimated solvent-accessible surface area for each residue is quite similar between the two softwares. This correlation validates the measurements obtained with the developed program.

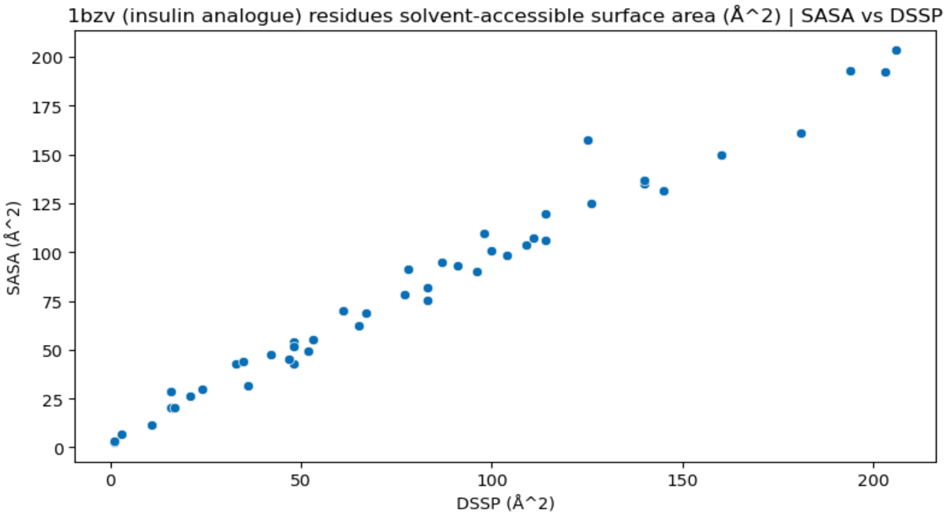


Figure 3: Solvent-accessible surface area of each residue of the protein 1bzv. Each point corresponds to one measurement of one residue. r (pearson correlation coefficient) = 0.989.

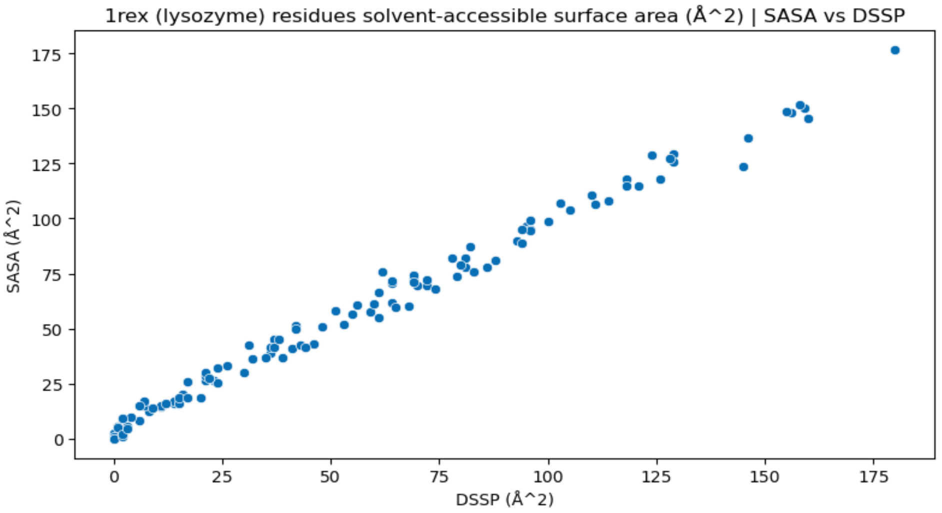


Figure 4: Solvent-accessible surface area of each residue of the protein 1rex. r = 0.995.

However, we can observe that in terms of the total accessibility surface of the protein, the difference between the obtained and expected results increases with the protein's size, as shown in Table 1. For a small protein (1bzv: 46 amino acids) the difference is 42.51 Å², contrary to a very big protein (1si4: 574 amino acids) where the difference is about 1959 Å².

Accessible surface (Å²)		
SASA	**DSSP**	
3749.31	**3706.8**	**1bzv**
6893.89	**6759.9**	**1rex**
27266.98	**25307.1**	**1si4**

Table 1: Accessible surface of protein (Å²) obtained with SASA vs DSSP.

3.2 Time of execution

The time of execution increased 2 to 3 times in the second version of the program (SASA v.2.0). For the small protein of 46 amino acids (366 atoms) the execution time passed from 8 to 4 seconds, and for the big protein of 574 amino acids (4379 atoms), it passed from 18 to 7 minutes.
It is important to note that the program was executed in a modeste machine with 4 CPUs, in more

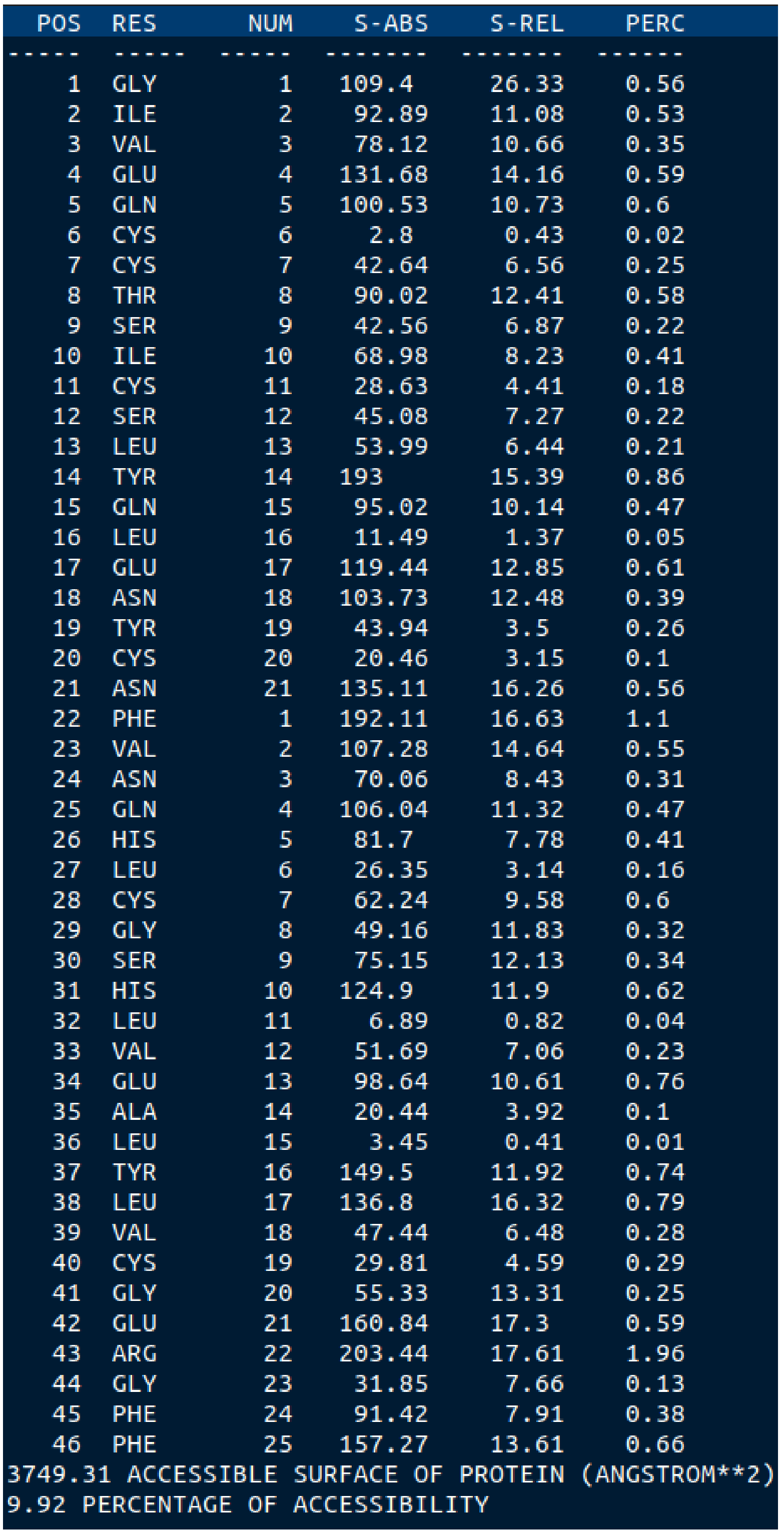
performant machines the time of execution is shorter.

REFERENCES

1. Accessible surface area. *Wikipedia* (2022).
2. Shrake, A. & Rupley, J. A. Environment and exposure to solvent of protein atoms. Lysozyme and insulin. *J Mol Biol* **79**, 351–371 (1973).
3. Lee, B. & Richards, F. M. The interpretation of protein structures: Estimation of static accessibility. *Journal of Molecular Biology* **55**, 379-IN4 (1971).
4. DSSP. https://swift.cmbi.umcn.nl/gv/dssp/.

ANNEX

python3 bin/SASA.py -pdb data/pdb/1bzv.pdb -n 92



```
  POS   RES       NUM     S-ABS     S-REL     PERC
-----  -----  -----  -------  -------  ------
    1  GLY        1    109.4     26.33     0.56
    2  ILE        2     92.89    11.08     0.53
    3  VAL        3     78.12    10.66     0.35
    4  GLU        4    131.68    14.16     0.59
    5  GLN        5    100.53    10.73     0.6
    6  CYS        6      2.8      0.43     0.02
    7  CYS        7     42.64     6.56     0.25
    8  THR        8     90.02    12.41     0.58
    9  SER        9     42.56     6.87     0.22
   10  ILE       10     68.98     8.23     0.41
   11  CYS       11     28.63     4.41     0.18
   12  SER       12     45.08     7.27     0.22
   13  LEU       13     53.99     6.44     0.21
   14  TYR       14    193       15.39     0.86
   15  GLN       15     95.02    10.14     0.47
   16  LEU       16     11.49     1.37     0.05
   17  GLU       17    119.44    12.85     0.61
   18  ASN       18    103.73    12.48     0.39
   19  TYR       19     43.94     3.5      0.26
   20  CYS       20     20.46     3.15     0.1
   21  ASN       21    135.11    16.26     0.56
   22  PHE        1    192.11    16.63     1.1
   23  VAL        2    107.28    14.64     0.55
   24  ASN        3     70.06     8.43     0.31
   25  GLN        4    106.04    11.32     0.47
   26  HIS        5     81.7      7.78     0.41
   27  LEU        6     26.35     3.14     0.16
   28  CYS        7     62.24     9.58     0.6
   29  GLY        8     49.16    11.83     0.32
   30  SER        9     75.15    12.13     0.34
   31  HIS       10    124.9     11.9      0.62
   32  LEU       11      6.89     0.82     0.04
   33  VAL       12     51.69     7.06     0.23
   34  GLU       13     98.64    10.61     0.76
   35  ALA       14     20.44     3.92     0.1
   36  LEU       15      3.45     0.41     0.01
   37  TYR       16    149.5     11.92     0.74
   38  LEU       17    136.8     16.32     0.79
   39  VAL       18     47.44     6.48     0.28
   40  CYS       19     29.81     4.59     0.29
   41  GLY       20     55.33    13.31     0.25
   42  GLU       21    160.84    17.3      0.59
   43  ARG       22    203.44    17.61     1.96
   44  GLY       23     31.85     7.66     0.13
   45  PHE       24     91.42     7.91     0.38
   46  PHE       25    157.27    13.61     0.66
3749.31 ACCESSIBLE SURFACE OF PROTEIN (ANGSTROM**2)
9.92 PERCENTAGE OF ACCESSIBILITY
```

Annex 1: 1bzv (insulin analogue) results obtained with SASA program. S-ABS : Accessible surface area (ASA), S-REL : Relative accessible surface area, PERC : Percentage of accessibility.