SOLVANT ACCESSIBLE SURFACE AREA CALCULATOR

Jorge Martínez García

Scientific Software and Data Carpentry

MSc in Bioinformatics

2024-2025

1. BACKGROUND

The solvent accessible surface area (SASA) of proteins is widely recognized as a critical factor in the study of protein folding and stability. SASA is described as the area around a protein defined by the centre of a hypothetical solvent sphere that corresponds to the van der Waals contact surface of the molecule. Amino acid residues within a protein can be categorized as either buried or exposed based on their SASA values. SASAs are categorized in various ways, from relative to absolute surface areas. It is not feasible to directly measure the precise SASAs of folded proteins at the atomic level through experimental methods (1).

However, the SASA of a native protein can be computationally estimated using its atomic coordinates. One method is the Shrake–Rupley algorithm, which employs a numerical approach by generating a mesh of points uniformly spaced around each molecule's atoms. This method assesses the surface area by counting the number of these points that are solvent-accessible. The points are set at a distance corresponding to the radius of a water molecule beyond the van der Waals radius of the atoms, akin to 'rolling a ball' across the surface. Each point is then checked against the surfaces of neighboring atoms to determine if it is buried or exposed. The total number of accessible points is then multiplied by the surface area that each point represents to calculate the ASA (2).

2. MATERIALS AND METHODS

In order to determine the solvent accessible surface area (SASA) of a target protein, its 3D structure was retrieved from its Protein Data Bank (PDB) file. To do this, the file was parsed, extracting the atom's coordinates while excluding the heteroatoms and water molecules.

For every atom in the protein structure, the absolute SASA was computed based on the surface area accessible to a solvent. This was quantified through the rolling probe method, where each atom was surrounded by a dynamically generated sphere using the Raff and Kuijlaars algorithm, which applies a quasi-uniform distribution of points across the sphere's surface. This sphere's radius incorporated the sum of the atom's van der Waals radius and the solvent probe radius. The construction of this sphere was performed by generating points across the sphere's surface as follows:

$$x = r\sin(\theta)\cos(\phi), \quad y = r\sin(\theta)\sin(\phi), \quad z = r\cos(\theta)$$

$$\theta = \cos^{-1}(1-2u), \quad \phi = 2\pi v, ; u, v \sim U[0,1]$$

where r is the radius of the sphere. The latitude θ and longitude Φ are determined stochastically, aiding in the even distribution of points over the spherical surface. The algorithm is specifically

chosen for its efficiency in distributing points to minimize areas of over- or under-sampling, which is critical for accurate surface area calculation.

Once the sphere is generated, the model computes the distance between each of its points and all neighbouring atoms within the sum of their van der Waals radii plus the probe radius. Points are considered as accessible if no neighbouring atoms' van der Waals sphere intersects with them.

The accessibility of each atom is the quantified by counting the number of accessible points, providing a direct measure of the solvent-accessible surface area. Then, the absolute SASA for each atom is then calculated based on the proportion of accessible points, using the formula:

$$SASA_i = 4\pi(r_i + r_{probe})^2\left(\frac{accessible\,points_i}{total\,points_i}\right)$$

where r_i is the atom's van der Waals radius and probe is the solvent probe radius (1.4 Å for water).

Following the calculation of absolute SASA for all atoms, the relative SASA at residue level was computed. The accessibility was normalised against a reference state, derived from a Gly-X-Gly conformation for each residue type, where X represents the residue in question. This method facilitates the comparison across different proteins or within different regions of the same protein (3).

3. RESULTS

The SASA of a set of 13 proteins was calculated using the framework, giving the absolute and relative SASAs found in Table 1

Table 1. Absolute and relative SASA values calculated for proteins from their PDB file

PROTEIN ID	ABSOLUTE SASA (Å²)	RELATIVE SASA(Å²)	PROTEIN ID	ABSOLUTE SASA (Å²)	RELATIVE SASA (Å²)
1a5d	17221.89	8712.35	2lyz	6692.97	3647.83
1b0b	6883.62	4077.36	2vta	14088.89	7212.89
1f88	28669.05	14194.22	4gcr	8870.93	4515.40
1kxk	12725.68	0	5p21	8384.93	4248.70
1l2y	1870.33	1097.49	6fqf	24092.77	12995.51
1stp	6881.90	4085.00	6moj	20920.29	11392.97
1tup	33401.08	14750.69			

We could observe that the absolute and relative SASA was successfully computed, except for the relative SASA of 1KXK. This was due to the undefined atoms present in this structure.

3.1. Comparison with NACCESS

These results were compared to the values obtained using NACCESS, a widely recognized tool for determining solvent accessibility. This comparison was aimed at validating our computational approach and providing a benchmark for the accuracy of our calculations. The comparison was conducted by calculating the percentage error between the SASA values from both methods, resulting in a 1.76% ± 1.11 of average error, with a maximal error of 4.17% and a minimal error of 0.66%. These statistics suggest that our approach performs well within acceptable limits for scientific computations, particularly in the context of complex biological models where slight deviations are common due to various modelling settings, like the number of points of the sphere or different algorithm.

3.2. Multiprocessing

To improve the performance of the application, we implemented a parallel processing version. This modification aimed to leverage the computational power of multi-core processors, reducing the overall execution time of the code. We measured the execution times of both the original and the parallelised version under identical conditions to ensure a fair comparison. The speed up achieved by introducing multiprocessing was quantified by calculating the ratio of the execution time of the single-process code to that of the multiprocess code.

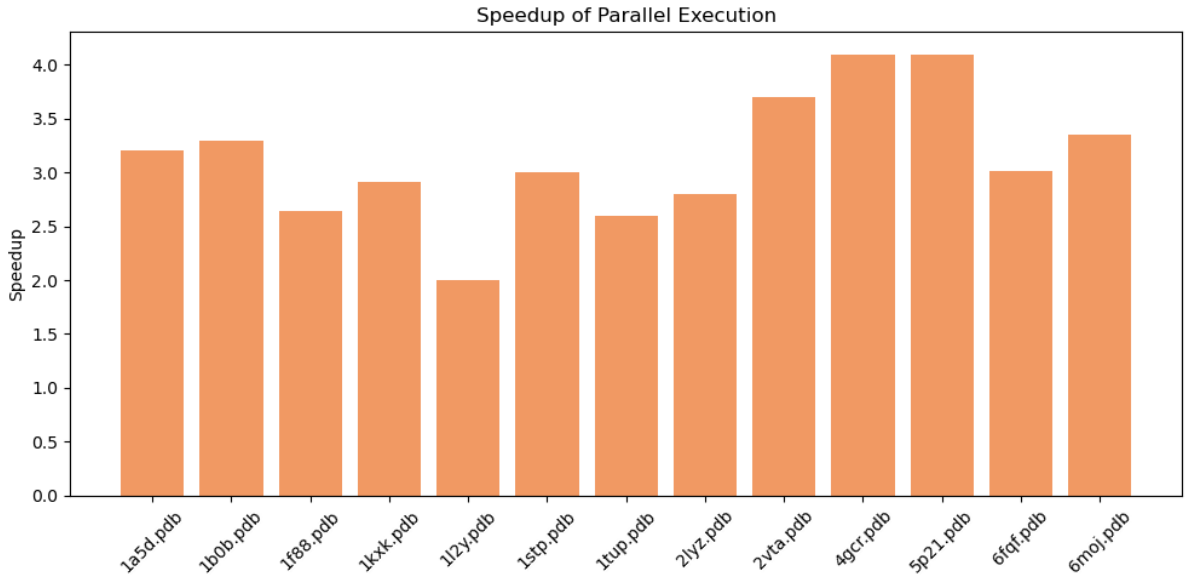

Figure 1. Performance achievements achieved by implementing multiprocessing across the various proteins

On average, the parallelised version of the code executed approximately three times faster than the single-threaded version (Figure 1). The maximum observed speedup was 4.1, while the minimum was 2.0. These results highlight the importance of multiprocessing, especially on the proteins that are larger and more complex, as they benefit from this approach.

4. CONCLUSION

Overall, our framework for the calculation of SASA, validated against NACCESS and optimised through multiprocessing, proves to be a reliable and efficient tool for protein analysis. The implementation of multiprocessing significantly reduced the execution time, enhancing computational efficiency, particularly for larger and more complex proteins.

This project not only showcases the effectiveness of our approach but also provided a valuable opportunity to enrich my Python programming skils.

5. REFERENCES

1. Ali SA, Hassan MI, Islam A, Ahmad F. A review of methods available to estimate solvent-accessible surface areas of soluble proteins in the folded and unfolded states. Curr Protein Pept Sci. 2014;15(5):456–76.
2. Shrake A, Rupley JA. Environment and exposure to solvent of protein atoms. Lysozyme and insulin. J Mol Biol. 1973 Sep 15;79(2):351–71.
3. Tien MZ, Meyer AG, Sydykova DK, Spielman SJ, Wilke CO. Maximum Allowed Solvent Accessibilites of Residues in Proteins. PLoS ONE. 2013 Nov 21;8(11):e80635.