**ECS 129 - Volume of a protein structure**

Authors - Wang Zach, Yidan Ma, Hanjing Shi

March 13, 2020

[**ECS 129 - Volume of a protein structure 1**](https://docs.google.com/document/d/1KVoWir4jEQcf3Qlpm19xKTglpNwKdq_BWAB5WwvGpCk/edit?ts=5e68550a#heading=h.pw1hqi1ikl9j)

[Introduction 2](https://docs.google.com/document/d/1KVoWir4jEQcf3Qlpm19xKTglpNwKdq_BWAB5WwvGpCk/edit?ts=5e68550a#heading=h.uxorbcmszw9y)

[Methods 3](https://docs.google.com/document/d/1KVoWir4jEQcf3Qlpm19xKTglpNwKdq_BWAB5WwvGpCk/edit?ts=5e68550a#heading=h.u1fks1t3u9qi)

[Results 6](https://docs.google.com/document/d/1KVoWir4jEQcf3Qlpm19xKTglpNwKdq_BWAB5WwvGpCk/edit?ts=5e68550a#heading=h.dmqghzxid2e4)

[Discussion 7](https://docs.google.com/document/d/1KVoWir4jEQcf3Qlpm19xKTglpNwKdq_BWAB5WwvGpCk/edit?ts=5e68550a#heading=h.ewb6y4qlgn7j)

[Bibliography 9](https://docs.google.com/document/d/1KVoWir4jEQcf3Qlpm19xKTglpNwKdq_BWAB5WwvGpCk/edit?ts=5e68550a#heading=h.flkvuoo78ei2)

**Introduction**

When it comes to the understanding of biology, we must understand the function of proteins. To understand the function of a protein, we must understand the geometry of it. The shape of protein is of significant importance as it corresponds to properties of the protein. Among the most important geometric measurements of proteins are volume and surface area, which are often necessary in order to determine the physical properties of proteins, including solubility and molecular weight, and biochemical properties of proteins, including intermolecular and intramolecular interactions. Adding all the properties up, an overall picture of a protein’s function can be quite precisely estimated. In this project, we focus mainly on computing the volume of a protein by adapting Monte Carlo techniques while also taking the surface area into consideration.

The volume of protein is challenging to compute as protein shapes are

considerably irregular. A lot of work has been done to tackle this problem previously. One of the earliest approaches is to add up the volume of each amino-acid residue (Zamyatnin, 1972). This is relatively easy to accomplish if the sequence of the protein is known, which is not always the case. However, the accuracy of this approach is low because neither the arrangement nor the packing of amino-acid residues is taken into consideration. As a result, it can only be used to calculate the compositional volume, an approximation of the actual volume. Another approach is to consider each atom in a macromolecule as a ball and the macromolecule as a union of balls, then compute the volume (or surface area) of this entity using the principle of inclusion-exclusion (Pavani and Ranghino, 1981). In other words, the volume of the union is expressed as an alternating sum of volumes of the common intersections of the balls. This method is considered to be accurate when no more than three balls intersect each other. Later on, this method was generalized to any number of simultaneous overlaps(Petitjean, 1994; Gibson and Scheraga, 1987). However, if forced to implement, the runtime of this algorithm would increase exponentially and additional analytical tools with high implementation complexity are necessary (Mach and Koehl, 2011).

In this project, we use a numerical and probability-based approach, named Monte Carlo technique, to tackle this problem. In comparison with other methods, Monte Carlo technique has surprisingly high accuracy and low implementation complexity.

**Methods**

We use Monte Carlo method to compute the volume of the input protein that is given as an input file as “Volume.crd”. In the file, the first row has a number for which we use M to denote the total number of data that follows. Each row has four values that describe the properties of a ball: x coordinate, y coordinate, z coordinate and the radius. In addition to the parameter M (2775 in this project), the second parameter is N, denoting the total number of random points we have used to calculate the probability that a random point will fall inside the union of the all balls.

**Procedure**

1. Build the box that contains the union of all balls
2. Generate N random points that are inside the box
3. Calculate the probability p that these points are inside the union of the all balls
4. Calculate the volume by multiplying the probability p and the volume of the box

**Output**

The program will generate an output file called “V\_N.txt” which contains six fields: Ve, σ, Ve-min, Ve-max, T and N. A sample row in the output file will be like the following without the heading:

Ve             σ             Ve-min   Ve-max T N

44596.08 39887.94 4708.13  84484.02 497  5

1. **Ve** stands for the the estimated Volume
2. **σ** stands for the standard deviation
3. **Ve-min** is the lower bound of the estimated Volume Ve
4. **Ve-max** is the upper bound of the estimated Volume Ve
5. **T** is the computing time, in microseconds, corresponding to the N
6. **N** is the number of random points

**Brief analysis**

The main program main.cpp reads a set of M atoms, and generates N random points inside the box that bound all these atoms. We are expecting to see how the

precision and accuracy of volume varies as N varies. Also, the time complexity of the method is expected to have big O(NM); the computing time is linear in the number of random points N and is linear in the number of atoms M.

To improve the time-consuming efficiency, we disallow unnecessary execution of loops. For example, when we check if a point is inside the box, we stop the loop as long as we find there is one point inside the box. Also, we would pass arguments by reference as possible as we can for every function. Since our project has a large input size, and parameter size, if we pass by value, the system will take more time to make a copy of the argument into the function parameter.  Passing by value would bring us an unnecessary performance penalty if we just need to perform read operation of the original parameters. Therefore, for reducing the computing time, we would choose the way that passes by reference for function arguments to avoid unnecessary copy of objects. Only when we need to modify the parameter will we then pass parameter by **Results**

Presentation - the V(N) and T(N) diagramA close up of a white wall

Description automatically generated

A picture containing wall, sky

Description automatically generated

**Analysis**

The V(N) diagram shows us the precision and accuracy of our estimated results plus standard deviation in a graphical way. At the beginning, the results are neither accurate nor precise when the N is very small. However, as N becomes sufficiently large, the V values become more precise and accurate until it gets stabilized, close to the actual volume 35490.34 Å3 . And the T(N) diagram suggests that the total computing time is indeed linearly proportional to the number of random points N.

**Discussion**

As shown in previous sections, we have successfully implemented the Monte Carlo techniques to computing the volume of a protein from a numerical and probability-based approach. This approach presents high accuracy and low implementation complexity. Thus, we decided to try to adapt it to compute the surface area of proteins by using a Monte Carlo-like function in our code, but it turns out that the same technique cannot produce good results for surface area. Potential reasons for this result will be explained later in this section. There are some other analytical solutions we have found in the literature.

From the academic literature, there are multiple concepts of macromolecule surface area. The most common and practically useful ones are the van der Waal surface area (vdWSA), which can be viewed as the total exposed surface area, and the solvent accessible surface area (SASA), which is defined as the surface traced by the center of a sphere rolled over the vdWSA of the macromolecule (Weiser, Shenkin, and Still, 1999). The radius of the sphere is determined by the solvent (usually, the radius of a water molecule).

From our perspective, the vdWSA of a macromolecule (represented in our program as a union of balls) could be computed in two distinct ways. One is by using the principle of inclusion-exclusion. In more detail, we can calculate the total surface area of the all the balls and subtract all the area embedded within other balls (unaccessible surface area) from the total value, which can be expressed as an alternating sum of surface area of the common intersection of the subsets of the union (Pavani and Ranghino, 1981). The implementation and corresponding analytical tools were introduced in Mach, P. and Koehl, P. (2011).

The other possibility is to adapt the Monte Carlo techniques we used for protein volume computation to estimate the total exposed surface area. If we random N positions in the setted boundary and evaluate whether some of them sit exactly on the surface of the protein (distance equals to radius), the probability we receive would be infinitely close to 0. Therefore, we tried to accomplish it in another way. First, we calculated the surface area of each ball and summed them up to determine the total area. Second, a set was established by including all the possible positions of points on the surface of any individual ball (regardless of being on the exposed surface or embedded surface). Third, we randomly selected N points from the set and test if each of them is within the range of another ball. Finally, we calculate the exposed surface area of the protein: dividing the number of points that are not within the range of another ball by N and then multiply this fraction by the total area. However, our implementation did not work well. We believe this is due to the huge complexity of the second step, and the problem may be solved in later studies if we can successfully establish a representative and inclusive set of coordinates.

**Bibliography**

Gibson K.D. & Scheraga H.A.(1987). Exact calculation of the volume and surface area

of fused hard-sphere molecules with unequal atomic radii. *Molecular Physics*.

62:5, 1247-1265, DOI: [10.1080/00268978700102951](https://doi.org/10.1080/00268978700102951)

Mach, P. and Koehl, P. (2011). Geometric measures of large biomolecules: Surface,

volume, and pockets. *J.* *Comput. Chem.*, 32: 3023-3038. doi:[10.1002/jcc.21884](https://doi.org/10.1002/jcc.21884)

Pavani, R. and Ranghino, G.(1981). Protein Volume in Solution. *ComputChem*. 6: 133.

Petitjean, M. (1994). On the analytical calculation of van der Waals surfaces and

volumes: Some numerical aspects. *J. Comput. Chem.*, 15: 507-523.

doi:[10.1002/jcc.540150504](https://doi.org/10.1002/jcc.540150504)

Weiser, J., Shenkin, P.S. and Still, W.C. (1999), Optimization of Gaussian surface calculations

and extension to solvent‐accessible surface areas. J. Comput. Chem., 20: 688-703.

doi:[10.1002/(SICI)1096-987X(199905)20:7<688::AID-JCC4>3.0.CO;2-F](https://doi.org/10.1002/(SICI)1096-987X(199905)20:7%3C688::AID-JCC4%3E3.0.CO;2-F)

Zamyatnin AA.(1972). Protein volume in solution. Progress in Biophysics and Molecular

Biology. 1972 ;24:107-123. DOI: 10.1016/0079-6107(72)90005-3.