3D image and graph based Computation of Protein Surface

Aruna Ranganath[1,*], K.C. Shet[2] and N. Vidyavathi[1]

[1]NMAMIT, Nitte, Udupi (Dt)-574 110, Karnataka(st), INDIA
[2]Dept of CSE, NITK, Surathkal, post 575 025, Karnataka(st), INDIA

Abstract

The accessible surface of a macromolecule is a significant determinant of its action. The interaction between biomolecules or protein-ligand is dependent on their surfaces rather than their bulk properties. Identifying these local properties of bimolecular surfaces plays a vital role in the area of biomedicine. For example, identifying binding sites, docking etc. In this paper we describe an algorithm for computing the molecular surface of protein. The algorithm considers the 3D structure of the protein as a 3D image. The algorithm constructs a 3D graph corresponding to the size of 3D image data volume; the graph nodes correspond to image voxels.The idea is drawn from the cost minimization in a graph developed by Thedens and Fleagle[1]. The algorithm uses a Dynamic Programming Technique to avoid combinatorial explosion of the legal local surface.

1 Introduction

Computing the three dimensional structure of protein molecules is an important problem in molecular biology. Proteins are sequences of amino acids and the constituent amino acids are discovered for a number of proteins. Finding the structure of protein molecules is important because it is their structure that determines their functioning in living organisms. Experimental procedures for protein structure determination are complex, expensive and extremely slow. To overcome this problem, computational methods are being developed to determine the structure of a protein molecule from its amino acid sequence. These computational methods are compute-intensive and hence stand to benefit from parallel programming. One such computational method depends upon the molecular surface area of protein molecules.

The exposure of protein atoms to solvent can be obtained by calculating the surface area of atoms in contact with solvent molecules. Computationally, this contacted surface is defined as the van der Waals' envelope. An atom is accessible if van der Waals' envelope can be drawn around any part of a given protein atom. [2]

2 Previous work in this field

The first molecular surface computation algorithms were numerical in nature (i.e., they computed by sampling) Connolly [3], Greer [4]. Connolly [3] computes the sampled surface (also known as the dot-surface) by placing a probe tangent to either one atom, or two atoms, or three atoms and checking to see if it intersects any of the other neighboring atoms. If it does not and it is tangent to

(i) One atom, then a dot is placed at the point of tangency between the probe and that atom,

[*] Corresponding author, aruna_ranganath@yahoo.com

(ii) Two atoms, then a concave arc of dots connecting the two points of tangency is created,

(iii) Three atoms, then a concave spherical triangle of dots is created between the three points of tangency.

This generates a dot-representation of the entire surface.

The analytic computation of the molecular surface was also first done by Connolly [5], [6]. Here a molecular surface is represented by a collection of spherical and toroidal patches as follows:

- The molecular surface for the regions of a molecule where the probe is in contact with a single atom are modeled by convex spherical patches.
- The molecular surface for the regions of a molecule where the probe is in simultaneous contact with two atoms are modeled by saddle-shaped toroidal patches.
- The molecular surface for the regions where the probe is in simultaneous contact with three atoms are modeled by concave spherical triangular patches.

The issues of algorithmic complexity of these algorithms have begun to be addressed only recently. Let ‘n’ be the number of atoms in a molecule and let ‘k’ be the average number of neighboring atoms for an atom in the molecule. By neighboring, we mean the atoms that are near enough to affect probe placement on a particular atom. Yip and Elber [7] presented an algorithm for computation of the list of neighboring atoms that is linear in ‘n’. It is based on spatial subdivision by a global grid. Perrot et al. [8], [9] presented a O(kn) algorithm that generates an approximation to the solvent-accessible surface. In this approximation, every concave spherical triangular patch between three atoms is represented by a planar triangle with vertices at the centers of these three atoms. Saddle-shaped toroidal regions and convex spherical patches are ignored. In terms of sequential algorithmic complexity this is good; however some points remain unaddressed here. This algorithm is inherently sequential; as it always needs to start from some concave spherical triangular region of the molecule and from there it proceeds by adding an adjacent face at a time. Besides being hard to parallelize, it fails for the cases where the solvent-accessible surface folds back to intersect itself or where the molecule has two or more sub-parts connected by only two overlapping spheres. Also, it cannot generate the interior cavities of a molecule.

In computational geometry, the α-hull has been defined as a generalization of the convex hull of point-sets by Edelsbrunner, Kirkpatrick, and Seidel [10], [11]. For $\alpha > 0$, the α-hull of a set of points P in two-dimensions is defined to be the intersection of all closed complements of discs with radius ‘α’ that contain all points of P. If we generalize this notion of α-hulls over point-sets to the corresponding hulls over spheres of unequal radii in three-dimensions, we would get the molecular surface (along with the surface defining the interior cavities of the molecule). It has been shown in [11] that it is possible to compute the α -hulls from the Voronoi diagram of the points of P. For $\alpha = 1$, the α -hull over the set of points P is the same as their convex hull. Richards [12] had also suggested computing the molecular surface by computing a 3D Voronoi diagram first and then using its faces to determine which nearby atoms to consider.

Edelsbrunner and Miicke[10] extend the definition of α-hulls to points in three-dimensions. Here an α -shape over a set of points P has been defined to be the polytope that approximates the α -hull over P, by replacing circular arcs of the α -hull by straight edges and spherical caps by triangles. An α -shape of a set of points P is a subset of the Delaunay triangulation of P. Edelsbrunner in [13], extends the concept of α -shapes to deal with weighted points (i.e. spheres with possibly unequal and non-zero radii) in three-dimensions. An α-shape of a set of weighted points P_w is a subset of the regular triangulation of P_w. Since these methods involve computing the entire triangulation first and then culling away the parts that are not required, their complexity is $O(n^2)$ in time. This is worst-case optimal, since an α-shape in three-dimensions could have a complexity of $O(n^2)$.

3 Methodology

Our aim is to develop an algorithm by identifying surface of the protein structure.

Steps:

1. Draw a 3D graph corresponding to the size of 3D image data volume. The graph nodes represent the image voxels.
2. The surface identification is based on the 3D surface connectivity requirements.
3. The total cost of identifying the surface is the sum of all the nodes forming surface.

The cost of the surface is defined as:

$$S = \sum_{x=1}^{X} \sum_{y=1}^{Y} C(x, y, z(x, y))$$

- The connectivity constraint guarantees surface continuity in 3D. The parameter N represents the maximum allowed change in the z-co-ordinate of the surface along the unit distance in x and y directions. If N is small, the legal surface is stiff and the stiffness decreases with larger values of N.

$$\text{for all } x \in [1, X] \quad \text{AND} \quad y \in [1, Y] : \\ z(x, y) - z(x-1, y)| \leq N \quad \text{AND} \quad |z(x, y) - z(x, y-1)| \leq N$$

- Each internal node of the graph may have 4(2N+1) legal neighbors that have to be examined when constructing the 3D graph.

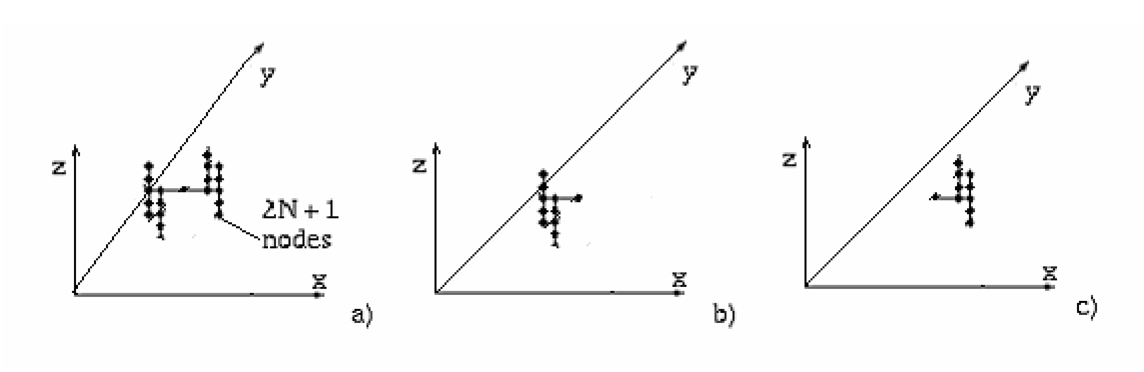

Figure 1: Neighbors of internal node: (a) Each internal node has 4(2N+1) neighbors. (b) Immediate predecessors (c) Immediate successors.

- The graph is searched starting from the vertical column with co-ordinates (1,1,z) in z -- x -- y co-ordinate order towards the column (X,Y,z).
- The cumulative surface cost is defined as the sum of the local cost associated with the node (x,y,z) and the sum of the two cost minima identified in the two columns constructed in the 3D graph that represent the immediate predecessors.

$$\begin{aligned} C_{cumulative}(x,y,z) &= C(x,y,z) \\ &+ \min\nolimits_{k\in[z-N,z+N]}(C_{cumulative}(x-1,y,k)) \\ &+ \min\nolimits_{k\in[z-N,z+N]}(C_{cumulative}(x,y-1,k)) \end{aligned}$$

- The surface construction proceeds in the reversed z-y-x order.
- Propagation of the connectivity constraint guarantees the legality of the resulting surface. The z-coordinate of the surface-node in the (x,y) column, denoted by D(x,y) is defined as

$$D(x,y) = z \text{ for which } C_{cumulative}(x,y,z) = \min_{k\in[zmin,zmax]}(C_{cumulative}(x,y,k))$$

where

$$\begin{aligned} zmax &= \min(Z, D(x+1,y)+N, D(x,y+1)+N) \\ zmin &= \max(1, D(x+1,y)-N, D(x,y+1)-N) \end{aligned}$$

- The backtracking process continues until the optimal node in the column (1, 1, z) is identified.

4 Algorithm

The methodology given above is more of abstract nature and suitable for all kinds of problems. When this methodology is applied to find the surface of the protein, a specific algorithm emerges. Hence we give below our novel algorithm to determine surface of a given protein.

1. Create a three dimensional matrix (X, Y, Z) corresponding to the size of the 3D image volume.
2. 3D graph construction.

   Starting from the column(1,1,Z) and proceeding in Z – X – Y co-ordinate order until the last matrix column (X,Y,z) is reached, calculate the costs of all graph nodes.

```
For Y=1 to y do
    For X=1 to x do
        For Z=1 to z do
            $C_{cumulative}(X,Y,Z) = C(X,Y,Z) + \min_{k€[z-N,z+N]}\{C_{cumulative}(x-1,y,k)\} + + \min_{k€[z-N,z+N]}\{C_{cumulative}(x,y-1,k)\}$
        End do
    End do
End do
```

3. Protein surface construction.

   Starting from the column (X, Y, z) and the proceeding in the reverse z – y – x co-ordinate order until the first matrix column (1, 1, z) is reached and considering the connectivity constraints, determine the minimum cumulative cost nodes defining the surface.

```
For x = X down to 1 do
    For y = Y down to 1 do
```

```
            $Zmax = \min (Z, D(x+1, y) + N, D(x, y+1) + N$
            $Zmin = \max (1, D(x+1, y) – N, D(x, y+1) – N$
            $D(x, y) = z$ which
            $C_{cumulative}(x, y, z) = \min_{k€[zmin,zmax]} (C_{cumulative}(x,y,k))$
        End do
    End do
```

5 Results

The following are the images of the surfaces generated by the algorithm. The 3D structures have been downloaded from the PDB database.

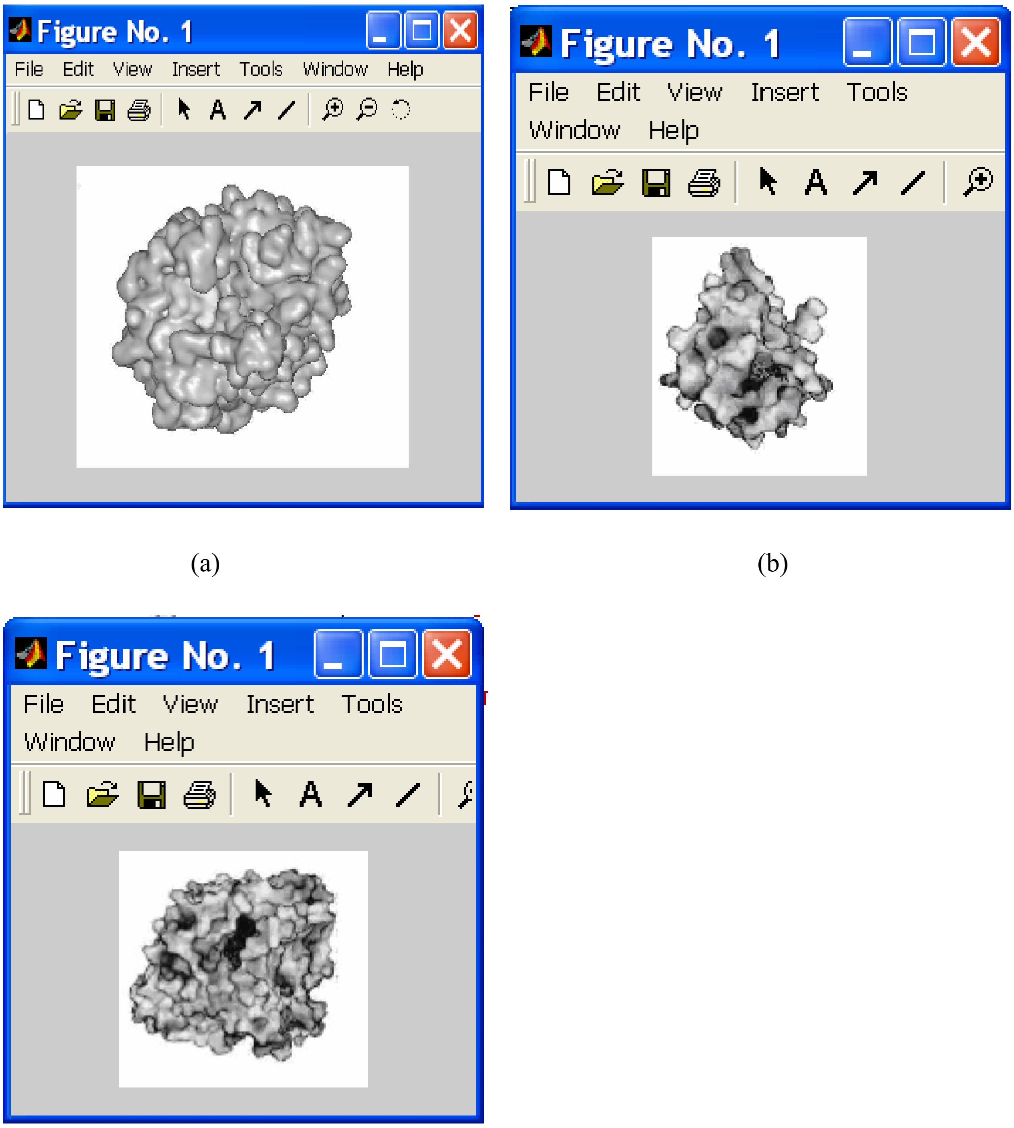
Figure No. 1
File Edit View Insert Tools Window Help
Figure No. 1
File Edit View Insert Tools Window Help
Figure No. 1
File Edit View Insert Tools Window Help

(a)

(b)

(c)

Figure 2: The surface of (a) CRAMBIN (b) Fleix and (C) Dihydropholate reductase generated by our algorithm.

It may be noted that this algorithm is unique, and it appears no one tried earlier. This has following features

(i) The time complexity is of the order O (n^3) due to 3D.

(ii) The space complexity is not an issue.

(iii) Accuracy and/or the smoothness appear to be better here as others contribute only dot-surface, whereas it is continuous-surface.

(iv) However, the smoothness needs to be quantified so that it can be optimized.

6 Conclusion

In this paper, we have investigated the structure of the protein molecule as a 3D image which is represented in the form of 3D graph. We have outlined the proposed algorithm to compute the protein surface using the graph theory techniques. At present we are not using any temporal information while generating the surface. Hence if the atoms move slightly from their positions we have to capture the new image and then recomputed the surface. We have ignored the physicochemical properties of the protein. The technical contribution of this paper is a combination of image processing and graph theory techniques. Our research in this direction is in progress for implementing the surface using wavelet representation.

7 References

[1] D. R. Thedens, D. J. Skorton, and S. R. Fleagle, “Methods of graph searching for border detection in image sequences with applications to cardiac MRI,” IEEE Trans. Med. Imag., vol. 14, pp. 42–55, Mar. 1995.

[2] http://www.biohedron.com/

[3] M. L. Connolly. Master's thesis, University of California at Berkeley, Berkeley, USA, 1981.

[4] J. Greer and B. L. Bush. Macromolecular shape and surface maps by solvent exclusion. In Proceedings of the National Academy of Sciences USA, volume 75, No. 1, pages 303{307, 1978.

[5] M. L. Connolly. Analytical molecular surface calculation. Journal of Applied Crystallography, 16:548{558, 1983.

[6] M. L. Connolly. Solvent-accessible surfaces of proteins and nucleic acids. Science, 221(4612):709-713, 1983.44

[7] V. Yip and R. Elber. Calculations of a list of neighbors in molecular dynamics simulations. Journal of Computational Chemistry, 10(7):921{927, 1989.

[8] G. Perrot, B. Cheng, K. D. Gibson, J. Vila, K. A. Palmer, A. Nayeem, B. Maigret, and H. A. Scheraga. Mseed: A program for the rapid analytical determination of accessible surface areas and their derivatives. Journal of Computational Chemistry, 13(1):1-11, 1992.

[9] G. Perrot and B. Maigret. New determinations and simpli_ed representations of macromolecular surfaces. Journal of Molecular Graphics, 8:141{144, 1990.

[10] H. Edelsbrunner. Algorithms in Combinatorial Geometry, volume 10 of EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1987.

[11] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. IEEE Transactions on Information Theory, IT-29(4):551-559, 1983.

[12] F. M. Richards. Areas, volumes, packing and protein structure. Ann. Rev. Biophys. Bioengg., 6:151-176, 1977.

[13] H. Edelsbrunner. Weighted alpha shapes. Technical Report UILU-ENG-92-1740, Department of Computer Science, University of Illinois at Urbana-Champaign, 1992.

[14] Image Processing, Analysis, and Machine Vision by Milan Sonka, Vaclav Hlav and Roger Boyle ISBN 0-534-95393-X.