Mining Protein Surfaces

Xiong Wang
Department of Computer Science
California State University, Fullerton
Fullerton, CA 92834-6870
wang@ecs.fullerton.edu

Abstract

Given a finite set of discrete points in three dimensional Euclidean space R^3, the subset that forms its surface could be different when observed in different levels of details. In this paper, we introduce a notion called α-surface. We present an algorithm that extracts the α-surface from a finite set of points in R^3. We apply the algorithm to extracting the α-surfaces of proteins and discover patterns from these surface structures, using the pattern discovery algorithm we developed earlier. We then use these patterns to classify the proteins. Experimental results show the good performance of the proposed approach.

1 Introduction

Discovering frequently occurring patterns has been explored in many different domains, e.g. sequences [1], trees [6], semistructured data [7], three dimensional data [9]. Classification is also one of the major tasks of data mining [3].

Protein classification is a very important research topic [3, 4, 6]. Traditionally, proteins are classified according to their functions. However, recently, many approaches have been proposed to classify proteins according to their structures, e.g. sequences [6], secondary structures [6], and three dimensional structures [9]. Many of these methods complemented the traditional approach. In [8, 9], we developed an algorithm for discovering frequently occurring patterns in three dimensional data and applied it to protein classification. While we succeeded in classifying two families of proteins with high recall and precision, experimental results showed that it was difficult to extend the approach to classifying more than two families of proteins. One reason is that proteins are large molecules, typically with several hundreds or even thousands of atoms. Many of the substructures that occur frequently in multiple proteins are not specifically related to their functions.

Significant studies have shown that the structure of the surface of a protein relates more to the function of the protein. For example, in [5], the authors examined the reliability of surface comparisons in searching for active sites in proteins. They suggested that, the detection of a patch of surface on one protein that is similar to an active site in another may indicate similarities in enzymatic mechanisms in enzyme functions, and implicate a potential target for ligand/inhibitor design.

In this paper, we define α-surface of a finite set of points in three dimensional Euclidean space and present an algorithm for extracting α-surfaces from finite point sets. We apply the algorithm to extracting α-surfaces of proteins. We then employ the pattern discovery algorithm that we developed earlier to find frequently occurring patterns on the α-surfaces and use these patterns to classify the proteins. The rest of the paper is organized as follows. In Section 2, we define α-surface and describe the surface extracting algorithm. Section 3 discusses how the surface extracting algorithm and the pattern discovery algorithm are applied to protein classification. Section 4 presents some experimental results. Section 5 concludes the paper.

2 α-Surfaces

Our definition of α-surfaces is inspired by the definition of α-shapes, introduced by Edelsbrunner and Mücke [2].

Definition 2.1 *Given a point O in three dimensional Euclidean space R^3 and a real number α $(0 < \alpha < \infty)$, an α-ball is the set of points $B(O,\alpha) = \{P | P \in R^3$ and $||P - O|| < \alpha\}$, where $||P - O||$ is the Euclidean distance between P and O. A closed α-ball $\overline{B}(O,\alpha)$ is the α-ball $B(O,\alpha)$ plus its bounding sphere, i.e. $\overline{B}(O,\alpha) = \{P | P \in R^3$ and $||P - O|| \leq \alpha\}$.*

Definition 2.2 *Given a finite set $\mathcal{D}$ of discrete points in R^3 and a real number α $(0 < \alpha < \infty)$, the α-surface $\mathcal{S}$ of $\mathcal{D}$ is defined as $\mathcal{S} = \{P | P \in \mathcal{D}$ and $(\exists O \in R^3$ such that $B(O, \alpha) \cap \mathcal{D} = \emptyset$ and $P \in \overline{B}(O, \alpha))\}$. When $B(O, \alpha) \cap \mathcal{D} = \emptyset$ and $P \in \overline{B}(O, \alpha) \cap \mathcal{D}$, we say that α-ball $B(O, \alpha)$ touches P. $P \in \mathcal{S}$ is called a surface point with respect to α (simply a surface point when the context is clear).*

For simplicity, we illustrate the notion in two dimensional Euclidean space. Figure 1 shows the α-surface of a finite point set. Surface points are highlighted by solid bullets in the figure. Apparently, given the same set of points $\mathcal{D}$, with respect to different α's, the α-surfaces are different.

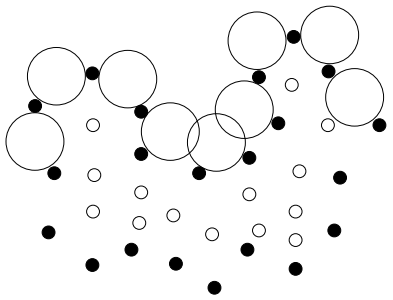

Figure 1. An α-surface.

The definition of α-surfaces is general. In the context of mining protein data, we need some adjustment. First of all, the surface of a protein is important to its function, because a protein reacts to its surrounding through its surface. Thus we are not concern with those parts of α-surfaces that are not *visible*, namely those surface points that are enclosed inside the proteins. Secondly, when α is small, the α-surface of $\mathcal{D}$ could be split to two pieces. A protein is one molecule. Its surface should be in one piece. We specify the adjustment in the following definition.

Definition 2.3 *Let α $(0 < \alpha < \infty)$ be a real number and $\mathcal{S}$ be the α-surface of a finite set $\mathcal{D}$. $\mathcal{S}$ is connected, if for any two surface points $P_1, P_2 \in \mathcal{S}$ there are a finite number of α-balls: $B(O_1, \alpha)$, $B(O_2, \alpha)$, ..., $B(O_n, \alpha)$, such that:*

(i) $B(O_i, \alpha) \cap \mathcal{D} = \emptyset$ $(1 \leq i \leq n)$.

(ii) $\overline{B}(O_i, \alpha) \cap \overline{B}(O_{i+1}, \alpha) \cap \mathcal{S} \neq \emptyset$ $(1 \leq i \leq n-1)$.

(iii) $P_1 \in \overline{B}(O_1, \alpha)$.

(vi) $P_2 \in \overline{B}(O_n, \alpha)$.

Notice that, (ii) requires two contiguous α-balls to touch at least one common surface point. Imagine that the α-ball is solid, so are the points in $\mathcal{D}$, and we roll the α-ball along the surface of $\mathcal{D}$. Intuitively, if an α-surface is connected, we can roll an α-ball from one surface point to another along the surface.

Starting from the point with the maximum X-coordinate in $\mathcal{D}$, the surface extracting algorithm rolls the α-ball to any surface point that can be touched in a breadth first manner[1]. The algorithm maintains a queue $\mathcal{Q}$ which holds a subset of the α-surface $\mathcal{S}$ that are under extension. The basic rolling procedure of the algorithm rolls the α-ball around one surface point in $\mathcal{Q}$, so that all its neighboring points in $\mathcal{S}$ will be touched at least once by the α-ball. These neighbors are added to $\mathcal{Q}$. Figure 2 illustrates the procedure. The α-ball is rolled around P_0 so that P_0's neighbors P_1, P_2, P_3, P_4, P_5, and P_6 are touched by the α-ball.

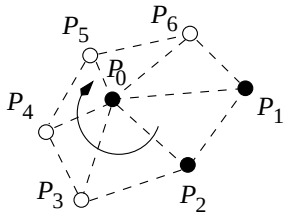

Figure 2. Rolling the α-ball.

Since the neighboring surface points are within distance 2α of the current surface point, to speed up the process, we partition $\mathcal{D}$ at the very beginning. Let x_{min} (x_{max}) be the minimum (maximum) X coordinate of all the points in $\mathcal{D}$, respectively. Let x_0, x_1, ..., x_n be defined as the following:

(i) $x_0 = x_{min}$,

(ii) $x_{i+1} = x_i + 2\alpha$ $(0 \leq i \leq n-1)$, and

(iii) $x_{n-1} \leq x_{max}$ and $x_{max} < x_n$.

We cut the range $[x_{min}, x_{max}]$ to segments $[x_i, x_{i+1})$ $(0 \leq i \leq n-1)$ with length 2α. Similarly, let y_{min} (y_{max}) be the minimum (maximum) Y coordinate and z_{min} (z_{max}) be the minimum (maximum) Z coordinate, respectively. We cut the ranges $[y_{min}, y_{max}]$ and $[z_{min}, z_{max}]$ to segments with length 2α. Each partition $Pt_{i,j,k}$ is a cube $Pt_{i,j,k} = \{(x, y, z) | x_i \leq x < x_{i+1},\ y_j \leq y < y_{j+1},\ and\ z_k \leq z < z_{k+1}\}$. Figure 3 shows a two dimensional example.

[1] Obviously, the point with the maximum X-coordinate in $\mathcal{D}$ is a surface point with respect to any α.

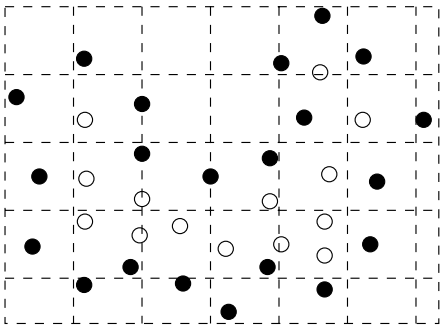

Figure 3. Partitioning the points.

For any given point $P = (x, y, z) \in \mathcal{D}$, let $i = \lfloor \frac{x - x_{min}}{2\alpha} \rfloor$, $j = \lfloor \frac{y - y_{min}}{2\alpha} \rfloor$, and $k = \lfloor \frac{z - z_{min}}{2\alpha} \rfloor$. P belongs to partition $Pt_{i,j,k}$ and the points that are within distance 2α of P are all located in the 27 partitions surrounding $Pt_{i,j,k}$.

Assuming that the points in $\mathcal{D}$ are evenly distributed, the complexity of the surface extracting algorithm is $O(\frac{|\mathcal{D}|^2}{N})$, where $|\mathcal{D}|$ is the size of $\mathcal{D}$ and $N = \lceil \frac{x_{max} - x_{min}}{2\alpha} \rceil \times \lceil \frac{y_{max} - y_{min}}{2\alpha} \rceil \times \lceil \frac{z_{max} - z_{min}}{2\alpha} \rceil$ is the total number of partitions.

3 Classifying Proteins

To evaluate the performance of the surface extracting algorithm, we applied it to classifying three families of proteins. We first utilized the surface extracting algorithm to find the surfaces of the proteins in the training data. We then employed the pattern discovery algorithm we developed before to find frequently occurring patterns from these surfaces. Finally, we used these patterns to classify the proteins in the test data.

Let $\mathcal{S}$ be the surface points outputted by the surface extracting algorithm. We apply our pattern discovery algorithm to $\mathcal{S}$ as follows. For any point $P \in \mathcal{S}$, we consider P and its k-nearest neighbors in $\mathcal{S}$ as a substructure and attach a local coordinate system SF to P (see Figure 4). We hash the node-triplets from the substructure to a three dimensional hash table (see Figure 5). The hash bin address is determined by the lengths of the three edges of the triangle formed by the triplet. Stored in the hash bin are a protein identification number, a substructure number, and the local coordinate system SF. We then consider each substructure as a candidate pattern and rehash it to evaluate its number of occurrences in the hash table. In this phase, we again decompose the candidate pattern to triplets and utilize the lengths of the three edges to access the hash table. All the triplets that were stored in the accessed hash bin are recognized as matches and their local coordinate systems SF are recovered based on the global coordinate system that defines the candidate pattern. The triplet matches are augmented to larger substructure matches when their recovered local coordinate systems match each other. The interested readers are referred to [8, 9] for more details.

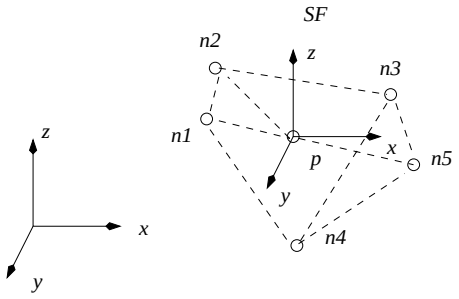

Figure 4. A substructure.

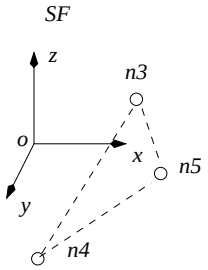

Figure 5. One of the triplets.

For each family i of the proteins, we identify two types of patterns on the surfaces of the training data, the *pro* patterns and the *con* patterns. The pro patterns occur more frequently in family i than in the other two families. The con patterns occur more frequently in the other two families than in family i. Each candidate pattern M found on the surfaces of the training data is associated with two weights pro^i and con^i where

$$pro^i = \frac{n_i - \max_{j \in \{1,2,3\} - \{i\}} \{n_j\}}{\max_{j \in \{1,2,3\} - \{i\}} \{n_j\} + 1}$$

$$con^i = \frac{\min_{j \in \{1,2,3\} - \{i\}} \{n_j\} - n_i}{n_i + 1}$$

Here n_i is M's occurrence number in the training data of family i. We add denominators to both weights

because we observed that some patterns are common to proteins from different families. Although they may still occur more frequently in some family, they really are not specific to any family. For each family we collect all the patterns having a weight greater than zero and use them as pro patterns and con patterns of that family, respectively. It can be proved that any pattern M that occurs in the training data is either a pro pattern or a con pattern of some family, unless M's occurrence numbers tie in all the three families.

We classify a test protein Q in the following way. Let $M_1^i, \ldots, M_{p_i}^i$ be all the pro patterns for family i. Family i obtains a pro score

$$\mathcal{N}_{pro}^i = \frac{\sum_{k=1}^{p_i} d_k \times pro_k^i}{\sum_{k=1}^{p_i} pro_k^i}$$

where

$$d_k = \begin{cases} 1 & \text{if } M_k^i \text{ occurs in } Q \\ 0 & \text{otherwise} \end{cases}$$

and pro_k^i is the weight associated with M_k^i. The protein Q is classified to the family i with maximum $\mathcal{N}_{pro}$. We add the denominator to make the score fair to all families. Notice that the maximum possible score for any family is 1. If we can not decide a winner from the pro scores, e.g. the scores are ties for two families, the con patterns are used to break the ties. Let $T_1^i, \ldots, T_{q_i}^i$ be all the con patterns for family i. Family i obtains a con score

$$\mathcal{N}_{con}^i = \frac{\sum_{k=1}^{q_i} d_k \times con_k^i}{\sum_{k=1}^{q_i} con_k^i}$$

where

$$d_k = \begin{cases} 1 & \text{if } T_k^i \text{ occurs in } Q \\ 0 & \text{otherwise} \end{cases}$$

and con_k^i is the weight associated with T_k^i. The protein Q is classified to the family i with minimum $\mathcal{N}_{con}$. If we still can not decide a winner, then the "no-opinion" verdict is given.

4 Experimental Results

We have implemented the surface extracting algorithm using C++ on a Sun Ultra 10 workstation running Solaris 8. We selected three families of proteins from SCOP [4]. SCOP is accessible at `http://scop.mrc-lmb.cam.ac.uk/scop/`. The three families pertain to Transmembrane Helical Fragments, Matrix Metalloproteases – catalytic domain, and Immunoglobulin – I set domains. In determining the structure of a protein, we consider only the C_α, C_β and N atoms. These atoms form the polypeptide chain backbone of a protein where the polypeptide chain is made up of residues linked together by peptide bonds. The peptide bonds have strong covalent bonding forces that make the polypeptide chain rigid. Figure 6 shows a protein whose PDB Code is 1cqr. It has 1089 atoms in the backbone. Figure 7 shows an α-surface found by the proposed algorithm, with respect to α=7.5. It has 242 atoms.

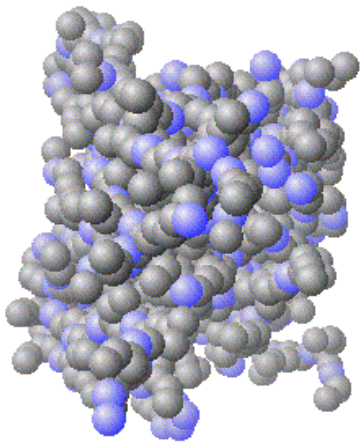

Figure 6. A protein (1cqr).

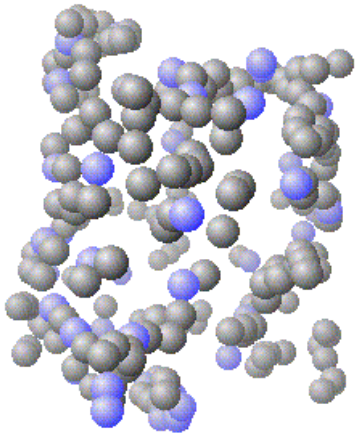

Figure 7. An α-surface of the protein (1cqr).

We classified the proteins as discussed in Section 3. When adjusting α in the surface extracting algorithm, we found that $\alpha = 7.5$ yielded the best result. When constructing substructures (patterns), we found the substructures with 6 points yielded the best result. In each of these substructures, there was a surface point together with its 5 nearest neighbors on the α-surface. The algorithm produced a set of surface points that were on average 25% of the size of the protein.

We use recall ($\mathcal{R}$) and precision ($\mathcal{P}$) to evaluate the effectiveness of our classification algorithm. Recall is defined as

$$\mathcal{R} = \frac{\mathcal{T} - \sum_{i=1}^{3} \mathcal{M}^i}{\mathcal{T}} \times 100\%$$

where $\mathcal{T}$ is the total number of test proteins and $\mathcal{M}^i$ is the number of test proteins that belong to family i but are not assigned to family i by our algorithm (they are either assigned to family j, $j \neq i$, or they receive the "no-opinion" verdict). Precision is defined as

$$\mathcal{P} = \frac{\mathcal{T} - \sum_{i=1}^{3} \mathcal{G}^i}{\mathcal{T}} \times 100\%$$

where $\mathcal{G}^i$ is the number of test proteins that do not belong to family i but are assigned by our algorithm to family i. With the 10-way cross validation scheme[2], the average $\mathcal{R}$ over the ten tests was 93.7% and the average $\mathcal{P}$ was 95.2%. It was found that 4.3% test proteins on average received the "no-opinion" verdict during the classification.

5 Conclusion

We have given a formal definition of surface points of a finite point set in R^3 and presented an algorithm for extracting such surface points. We applied this algorithm, together with previously developed algorithms for 3D pattern discovery, to classifying three families of proteins. In our previous work [9], we also tried to classify three families of proteins and the recall and precision dropped to 80%. The results reported here are much better. The idea of *pro* patterns and *con* patterns can be extended to more than three families. For our future work, we will conduct comprehensive experiments on more protein families to find interesting patterns on their surfaces and to classify them. We will also extend our algorithm to applications in three dimensional visualization.

6 Acknowledgments

The author thanks Dr. Jason T. L. Wang for his comments and Dr. Jane Richardson for helpful discussion during the Atlantic Symposium on Computational Biology, Genome Information Systems & Technology.

[2] That is, each family was divided into 10 groups of roughly equal size. Ten tests were conducted. In each test, a group was taken from a family and used as test data; the other nine groups were used as training data for that family.

References

[1] R. C. Agarwal, C. C. Aggarwal, and V. V. V. Prasad, "Depth first generation of long patterns," *Proceedings of the 6th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 108–118, Boston, Massachusetts, 2000.

[2] H. Edelsbrunner and E. P. Mücke. "Three-Dimensional Alpha Shapes," *ACM Transactions on Graphics*, 13(1):43–72, 1994.

[3] R. King, A. Karwath, A. Clare, and L. Dephaspe, "Genome scale prediction of protein functional class from sequence using data mining," *Proceedings of the 6th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 384–389, Boston, Massachusetts, 2000.

[4] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. "SCOP: a structural classification of proteins database for the investigation of sequences and structures," *J. Mol. Biol.*, 247:536–540, 1995.

[5] M. Rosen, S. L. Lin, H. Wolfson, and R. Nussinov. "Molecular shape comparisons in searches for active sites and functional similarity," *Protein Engineering*, 11:263–277, 1999.

[6] J. T. L. Wang, B. A. Shapiro and D. Shasha, editors. *Pattern Discovery in Biomolecular Data: Tools, Techniques and Applications*, Oxford University Press, New York, 1999.

[7] K. Wang and H. Liu, "Discovering Structural Association of Semistructured Data," *IEEE Transactions on Knowledge and Data Engineering*, 12(3): 353–371, May/June 2000.

[8] X. Wang, J. T. L. Wang, D. Shasha, B. A. Shapiro, S. Dikshitulu, I. Rigoutsos, and K. Zhang. "Automated discovery of active motifs in three dimensional molecules," *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 89–95, Newport Beach, California, 1997.

[9] X. Wang, J. T. L. Wang, D. Shasha, B. A. Shapiro, I. Rigoutsos, and K. Zhang. "Finding Patterns in Three Dimensional Graphs: Algorithms and Applications to Scientific Data Mining," Accepted to *IEEE Transactions on Knowledge and Data Engineering*, 2001.