Exploiting Geometrical Properties on Protein Similarity Search

Saiful Akbar 1, Josef Küng 2, Roland Wagner 3
FAW, Johannes Kepler University of Linz, AUSTRIA
{1 saiful.akbar, 2 jkueng, 3 rrwagner}@faw.uni-linz.ac.at

Abstract
This paper discusses about several combinations of protein similarity measurement-methods, with respect to normalization, spatial partitions, geometrical properties, and distance metrics. We compare the effectiveness of possible combinations to each other. Our experiment shows that the feature based on fractional occupancy outperforms other methods. In addition, merging individual features might also yield good result. A prototype of 3D protein geometrical-similarity retrieval system is built for implementing our approach.

1. Introduction
Similarity searching on the three-dimensional structure of protein molecules is challenging problem. The search is needed due to such a tool can aid many scientists in the identification of new type of protein architecture or in the development of procedure for drug design. It also can help to discover unexpected evolutionary and functional inter-relations between proteins. X-Ray Crystallography and Nuclear Magnetic Resonance (NMR) are de facto standard tools for observing protein structures. When protein structure could be predicted, the next step is looking for similarity of observed protein against those in protein databank. By observing the similarity, biological scientist will be able to classify proteins, which have common ancestry, investigating if they are already in the databank, or if they are a mutation of available proteins, which are already priory known.

Generally, measuring similarity and classifying proteins in a database require experts that have deep knowledge of molecular biology domain. This is due to the measuring methods, such as SCOP, DALI, CATH, and FSSP, usually base on a particular biological conception of structural similarity of proteins. In addition, most of the similarity searches related to protein structure, such as [1] and [2], are sequence-alignment searches. They yield good accuracy of the searching result, but are very time consuming.

Another approach to model structural similarity of protein is the geometrical approach. An experiment [3] suggested that nearly all equivalent patches larger than 10 atoms have an identical geometry. Thus, geometry provides high significance for similarity search in protein database. Some researchers, such as [4] and [5], show that approaching similarity of protein by its geometry is promising. In this paper, we develop a similarity-searching tool by exploiting some geometrical properties of the proteins. Note that, such a tool will be useful for filtering or finding a set of selected candidates from large database, for the next more complex structural similarity observation.

The rest of this paper is organized as the following. In Section 2 we describe some related works done by previous researchers, and how this paper distinguishes itself from that. Section 3 describes several possible combinations for exploiting geometrical properties of proteins. In Section 4 we present the implementation of our prototype, and Section 5 we discuss the experimental result and some highlighted observations. Section 6 concludes the paper.

2. Related Work
A nearest neighbor classification of proteins based on geometry is proposed by [4]. The idea behind the proposed approach is providing a first filter for a fast selection of candidate structures from a large database. It addresses three important issues. First, it extracts geometry of the object and maps it into 3D shape histogram, using three options of 3D-decomposition technique: ball (shell), sector and combination of both. Second, it employs quadratic distance in order to counter small shift and rotation problem of the objects. Third, it optimizes query processing by employing a filter-refinement architecture, which provides a multi-step k-nearest neighbor search. The method also combines thematic and shape histogram in order to obtain better result. It extracts histograms from uniformly distributed surface points taken from the molecular surfaces. As long as we know, it employs only the histogram of atoms, instead of exploiting some possible geometrical properties of the objects.

Another geometrical approach of protein similarity measurement is proposed by [5]. It counters rotational variations of protein structures by employing Light Field Descriptor. Each model is rotated several times in order to obtain its projection image from some certain camera position. Features are defined by observing several features of 2D shape and contour of projection images, i.e Zernike moments, Fourier coefficients, eccentricity, and circularity. Combining the features, it calculates the distance of features, using L1 norm, to get the
dissimilarity of two proteins. Note that, this method needs to generate an intermediate representation, i.e. 2D images of 3D protein object, before extracting the features.

This paper distinguishes itself from the previous work in the following ways:

1. Our approach intentionally exploits several methods of quantizing 3D representation of protein object to several possible partitions.

2. It also exploits geometrical properties of protein structure without considering the functionality of the protein itself such as what is done in a sequence alignment.

3. It compares some possible combinations of spatial partition schemas, geometrical properties, and distance metrics; and investigates them to have the most effective combination. We also investigate which combinations might be complementing to each other.

3. Exploiting Geometrical Properties

In order to measure protein similarity, or in general, the geometry of 3D objects, we have to address several issues, i.e.: normalization, spatial partitions, geometrical features, and metrics. This section will discuss the issues and some considerations in detail.

3.1 Normalization

In order to compare the similarity among 3D objects, first we have to transform the objects into their canonical representation with respect to translation, scaling, rotation, and mirroring variance. However, in our opinion this is not always the case especially for protein similarity, due to the following reasons:

1. In the case of protein similarity, scaling invariance might be not desired [4] as the scale itself may contain information that we intend to compare.

2. Scaling and mirroring the objects are not needed if the similarity model inherently supports rotation and mirroring invariance, e.g. Fourier descriptor. They are also not required if the objects is stored in a standardized orientation [4].

3. Our preliminary experiment shows that in a certain similarity model, which is variant to rotation and mirroring, rotating and or mirroring to an object might yield an inappropriate pose.

Therefore, in our experiment, we intentionally provide the possibility of having either semi-normalized or fully normalized pose of objects. The semi-normalized pose is obtained by transforming them such that their center of mass is in the origin and scaling them to a certain unit of bounding box. Full-normalized pose is aimed to preserve the objects invariant to translation, scaling, rotation, and mirroring. Translation and scaling invariance are obtained by the same way as in the semi-normalized form. Rotation invariance is obtained by the following way. First, employ Principal Component Analysis (PCA) to the objects in order to get the principal axis. Second, rotate them such that the first major axis is adjacent to x-axis, the second to y-axis, and the third to z-axis. Mirroring invariance is obtained by flipping the objects such that the larger part is on the positive side.

3.2 Spatial Partitions

There exist four basic space-partitioning methods: Shell (Ball), 3D Grids, Spherical Map, and Sector, as illustrated in Fig. 1.

1. Shell (BALL)
   This partition is obtained by decomposing 3D space into concentric shells around the center point such that the partition volume is growing from the smallest to the biggest with same and fixed size of growing radii. Note, that this partition inherently supports rotational invariance of the objects.

2. 3D Grid (GRID)
   3D Grid partition is obtained by quantizing the minimum bounding cube of the 3D object into smaller fixed unit cube.

3. Spherical Map (SPH)
   A spherical map is a partition of the sphere's surface into three kinds of elements: vertices, edge, and faces. A 3D partition is built by connecting vertices of a face to the sphere's center. Using this method, we intend to partition a 3D space into different volumes. Partitions closed to the equator line are bigger than partitions, which are closed to the two poles. However, it is obvious that these partitions will be invariant to scaling.

4. Sector/Polyhedron (ICO)
   Sector partition is done by quantizing 3D space into sectors such that the partitions will have the same area of spherical-surface. Practically, we can make use of vertices on the surface of regular polyhedron, such as dodecahedron or icosahedron; and emerge the cords from object center to the vertices. In our implementation, we make use of icosahedron. It is obvious that the partition will inherently support scaling invariance.
We might extend and combine the basic methods in order to have new partition methods, e.g. ellipsoid partition or box partition, whose radii or edges obtained by following 3 principal axes; cell-sector, or ellipsoid-sector. These extensions and combinations are beyond the scope of this paper.

3.3 Geometrical Properties

In this section, we list some geometrical properties we extract from the objects.

1. Fractional Occupancy (FO)
   Adapting from [6], we define fractional occupancy as the ratio of the number of atoms in a partition \((Na_i)\) to the volume of partition unit \((Vp_i)\).
   \[
   FO_i = \frac{Na_i}{Vp_i}
   \]  

2. The inverse of Local Elongation (LE) [8]
   By performing Principal Component Analysis on the atom points, we obtain eigenvalues \(\lambda_1\), \(\lambda_2\), and \(\lambda_3\) in decreasing order. Local Elongation is inversely related to \(\lambda_2/\lambda_1\).
   \[
   LE_i = \frac{\lambda_2}{\lambda_1}
   \]  

3. Object Bumpiness (BP) [8]
   The ratio \(\lambda_3/\lambda_1\) gives a measure of the bumpiness.
   \[
   BP_i = \frac{\lambda_3}{\lambda_1}
   \]  

4. Cords (CO) [6]
   A cord is a vector that goes from the centre of mass of an object (the origin) to a representing point in a partition. Let \(V_i\) be a vector emerging from the origin to the representing point, the feature related to this property is defined as the distance of the vector.
   \[
   CO_i = |V_i|
   \]  

5. Surface Curvature (CU) [8]
   Let \(p_i\) be a representing point in a partition \(i\), the surface curvature is defined as the curvature formed from \(p_i\) with other points in its partition neighbors. The curvature can be approximated by calculating the angles of triangles formed by \(p_i\) and the neighbor points, and subtracting the angles from the biggest possible angle in the sphere \(2\pi\). In this paper, without loosing the information we concern on the sum of angles \(\omega_j\) without subtract it from \(2\pi\).

\[
CU_i = \sum_j \omega_j
\]  

The first issue to be concerned with is the same as those of cords features (CO) above, and we decide to use the most distant point. The second is about feature normalization. As we calculate the total angle formed by \(p_i\) and its neighbors, the value may range from zero to \(2\pi\). We refer to this kind of feature as non normalized-feature. We cannot merge this feature with others without normalizing it in advance.

6. Center of Gravity (CG)
   Let \(V_i\) be a vector formed by the origin and the mean point in a partition volume, the feature in partition \(i\) with respect to the center of gravity is defined as:
   \[
   CG_i = |V_i|
   \]  

3.4 Metrics

Having geometrical features extracted from proteins, now the further task is having the distance metrics in order to calculate the dissimilarity between features. There are many methods for comparing features, such as Minkowski-form distance, \(X^2\) statistics, Quadratic-form distance, Match distance, Kosmogorov-Smirnov distance, and earth mover’s distance (EMD) [9]. In general, a distance metric differs from each other on how they define the dissimilarity and whether or not it takes into account cross-bin dissimilarity.

In our implementation, we select two metrics, which represent different approaches of measuring dissimilarity. The first, Minkowski \(L1\) norm, is a metric which does not consider cross-bin dissimilarity, while the second, Quadratic-form distance, is the opposite. The followings are detail description of the metrics:

1. Minkowski \(L1\) norm (L1)
   Minkowski \(L1\) norm is defined as
   \[
   D(P, Q) = \sum_{i=1}^{n} |p_i - q_i|
   \]  

   Note that we decide to use \(L1\) instead of \(L2\) due to our preliminary experiment, which has indicated that making use of \(L1\) gives better result than those of \(L2\).

2. Quadratic-form distance (QD)
   Let \(a_{ij}\) be a member of a similarity matrix \(A\) that provides cross-bin dissimilarity information, Quadratic-form distance is defined as
   \[
   D(P, Q) = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} (p_i - q_i)(p_j - q_j)}
   \]

   In our experiments, we make use of \(a_{ij} = 1 - d_{ij} / d_{max}\) as used by [9].

4. Implementation

We implement a geometry based 3D protein similarity retrieval system, which employs various combinations of methods we described in the previous section. Fig. 2
shows the searching result using `1njc.brk` as the query object. The image in the top left is the query object, while the images in the right side are the k-nearest searching result with k=25. In the bottom, the detail information and the 3D stick representation of a selected protein are shown.

![Fig. 2. An example of searching result with query object 1njc.brk, making use of the feature of Bumpiness](image)

For our experimental system, we use a dataset of 3D proteins downloaded from EMBL database [10]. We select 544 files, which are the members of big class, i.e. a class with 15-35 members. From a 3D protein object, we extract the features of dimension 128. In order to measure the effectiveness of the features on searching, we follow the classification in [5].

We have 2 different options for normalization method, 4 partition methods, 6 geometrical features, and 2 distance methods. Therefore, we must have 96 combinations. As some of combinations do not work properly, we have 72 combinations of methods from the basic methods we described in the previous section. In addition, we combine basic features in order to form a new feature, we called coarse feature (CA). Totally, we have 74 cases to compare to each other as shown in Table 1.

Coarse feature is obtained by combining the geometrical properties in respect with atom type partition. Atoms are partitioned based on their type, i.e. Carbon (C), Nitrogen (N), Oxygen (O), Hydrogen (H), and others. Thus, we extract the following geometrical properties: FO (dimension 1), LE (dimension 1), BP (dimension 1), Bounding Box (BB, dimension 3), CO (dimension 8), and CU (dimension 8). As we have whole protein (1 partition) and each type of atom (5 partitions), finally we have features of dimension 132.

While combining basic features we have to address several issues. The first is the difference of scale. For example, the maximum value of FO might be much greater than one, while the value of LE and BP ranges from zero to one. Hence, the features need to be normalized to the same scale. The second issue is the difference of dimension. It does not make any senses to extract 1 dimension of CO or CU. Therefore, we decide to have 8 dimensions of CO and CU as we consider it as the least possible dimension. However, in general, each feature describes only one type of information: cords and curvature, respectively. Hence, for each feature of dimension more than 1, we divide it with the number of dimension. The last issue is about the distance metrics. It is obvious that employing cross-bin dissimilarity metric between different features does not work, except for bins in the same feature such as CO or CU. Hence, we decided to employ only L1 norm for this case.

### Experimental Result and Discussion

In order to compare the effectiveness of the methods, we measure the average of First Tier (1\textsuperscript{st}), Second Tier (2\textsuperscript{nd}), and Nearest Neighbor (NN) [6], as depicted in Table 1. First Tier indicates the percentage of top k-1 matches, where k is the size of the class. Second Tier indicates the percentage of top 2(k-1) matches. Nearest Neighbor indicates the percentage of the first top match, excluding the query object itself. The higher score of them indicates that a method is more effective. Referring to Table 1, we underline the following important observations.

In general, our experimental result indicates that the geometrical features can effectively discriminate groups of 3D protein in a certain degree. Our experiments show that making use of full normalization does not always yield better result. Instead, on average semi-normalization method outperforms full-normalization method. Recall that semi-normalization method preserves only translation and scaling invariance, while full-normalization method preserves translation, scaling, rotation, and mirroring invariance. We found that in our case using L1 is better than QD. This could be due to the bin-similarity matrix $(A)$. Hence, another experiment using different bin-similarity matrix has to be done in order to have more general conclusion.

Comparing to other features, FO features are the most effective at discriminating groups of protein. However, its discriminating power is poor for some classes of proteins. Coarse feature (CA) is the best second below FO; and interestingly for some classes on which discriminating power of FO is poor, it yields higher accuracy. We observe that the 4 combinations: i.e. SEMI_BALL_FO, SEMI_GRID_BP, SEMI_SPH_CG, and SEMI_CA, occupy the most effective nearest neighbor searching on about 88% of available classes in our experimental dataset. Therefore, making use of them in a multi-channel similarity system might result in higher accuracy of nearest neighbor search.

Different from coarse feature (CA) which combines some individual features and forms a new feature, a multi-channel similarity system deals with a mechanism to promote a new searching result from prior results obtained by making use of the individual features. One possible mechanism is employing voting mechanism between prior results. Another experiment needs to be done to investigate this possibility.
6. Conclusion

In this paper, we gave experimental evidence that geometrical properties can be good features for protein similarity searching. The main contribution of this paper is the approach of exploiting some possible geometrical features of 3D protein, and making use of some possible combinations, with respect to normalization, spatial partitions, geometrical features, and distance metrics. In addition, the paper gives an early experience to find some possible complement features, which are prospectively usable for building a multi-channel 3D protein similarity system. Several research issues merit further investigation: Would other bin-similarity matrices improve the performance of using QD? How to take advantage of complementing features?

Table 1. Evaluation of all combined methods, using First Tier, Second Tier, and Nearest Neighbor

<table>
<thead>
<tr>
<th>Method</th>
<th>FULL</th>
<th>QD</th>
<th>SEMI</th>
<th>QD</th>
</tr>
</thead>
<tbody>
<tr>
<td>BALL</td>
<td>CG</td>
<td>35.5</td>
<td>44.0</td>
<td>71.7</td>
</tr>
<tr>
<td>GRID</td>
<td>BP</td>
<td>35.9</td>
<td>44.2</td>
<td>82.9</td>
</tr>
<tr>
<td>SPH</td>
<td>BP</td>
<td>25.1</td>
<td>33.7</td>
<td>71.5</td>
</tr>
<tr>
<td>ICO</td>
<td>BP</td>
<td>22.1</td>
<td>28.7</td>
<td>67.8</td>
</tr>
<tr>
<td>CA</td>
<td>48.6</td>
<td>57.3</td>
<td>84.7</td>
<td>49.3</td>
</tr>
</tbody>
</table>

Acknowledgement

This research is supported by ASEA-UNINET Technology Grant of OeAD.

References