This is an author produced version of a paper published in


This version may not include final proof corrections and does not include published layout or pagination.

Citation Details

Citation for the version of the work held in ‘OpenAIR@RGU’:


Citation for the publisher’s version:


Copyright

Items in ‘OpenAIR@RGU’, The Robert Gordon University Open Access Institutional Repository, are protected by copyright and intellectual property law. If you believe that any material held in ‘OpenAIR@RGU’ infringes copyright, please contact openair-help@rgu.ac.uk with details. The item will be removed from the repository while the claim is investigated.
Dimensionality Reduction in Patch-Signature Based Protein Structure Matching

Zi Huang
Xiaofang Zhou
Dawei Song
Peter Bruza

1School of Information Technology and Electrical Engineering
University of Queensland,
St. Lucia, QLD 4072
Australia
Email: {huang,zxf}@itee.uq.edu.au

2Distributed Systems Technology Centre
Level 7, General Purpose South
University of Queensland,
St. Lucia, QLD 4072
Australia
Email: {dsong,bruza}@dstc.edu.au

Abstract

Searching bio-chemical structures is becoming an important application domain of information retrieval. This paper introduces a protein structure matching problem and formulates it as an information retrieval problem. We first present a novel vector representation for protein structures, in which a protein structural region, formed by the vectors within the region, is defined as a patch and indexed by its patch signature. For a k-sized patch, its patch signature consists of 7k – 10 inter-atom distances which uniquely determine the patch’s spatial structure. A patch matching function is then defined. As structures for proteins are large and complex, it is computationally expensive to identify possible matching patches for a given protein against a large protein database. We propose to apply dimensionality reduction to the patch signatures and show how the two problems are adapted to fit each other. The Locality Preservation Projection (LPP) and Singular Value Decomposition (SVD) are chosen and tested for this purpose. Experimental results show that the dimensionality reduction improves the searching speed while maintaining acceptable precision and recall. From a more general point of view, this paper demonstrates that information retrieval techniques can play a crucial role in solving this biologically critical but computationally expensive problem.

Keywords: Protein Structure Matching, Similarity Measure, Dimensionality Reduction

1 Introduction

Information science has been applied to computational biology, resulting in a new field called Bioinformatics, which investigates “the collection, archiving, organization and interpretation of biological data” (Orengo, Jones & Thornton 2003).

Discovering functional relationships between proteins is recognized as a central task of modern bioinformatics. The problem of comparing amino acid sequences in proteins has been investigated extensively in the past. The research focus has now been shifted towards higher level biological structures and functions. It has been found that it is common for proteins that do not share significant sequence similarity to have significant structural similarity (thus potentially functional similarity) (Mount 2001). When the sequence similarity is below a certain percentage, say 20%, only structure analysis can reveal the potential relationship which may be hidden at the sequence level (Bourne & Weissig 2003). It is known that the protein’s unique three-dimensional structure often determines its properties. Finding proteins with similar substructures is an important problem, as certain structural regions of a protein often perform some specific functions, and having one or more similar 3D substructures has been considered as an essential condition for potential protein interaction.

As 3D protein structures are large and complex, it is computationally expensive to identify possible locations and sizes of the matching structural regions for a given protein against a large protein database. A commonly used structure representation is the inter-atom distance matrix. As the complexity of the distance matrix representation is quadratic to the number of atoms, it is very expensive for processing a large number of proteins.

To alleviate this problem, we introduce a patch signature model which has been recently proposed based on a vector representation for protein structures. A structural region is defined as a patch formed by the vectors within the region. The patch signature is used to characterize a patch. Compared to the traditional distance matrix representation, patch signature is more compact and linear to the number of atoms. The matching function between two patches is then defined as pair-wise comparisons between their patch signatures.

However, the matching stage can still be very expensive since the dimensionality of patch signature data can be large when the size of patch is large. A obvious solution to more efficient patch matching is to reduce the dimensionality of patch signatures while maximally preserve the matching function defined between two patches in the resultant lower dimensional space. Dimensionality reduction has been extensively applied in information retrieval. The goal is to find an “intrinsic” subspace, which is the best lower dimensional approximation of the original space depending on the objective function a dimensionality reduction algorithm tries to preserve. A well known approach is Singular Value Decomposi-
form regular secondary structures, including the turns. Different regions on the amino acid sequence 
atom construct the mainchain (or backbone) of an amino acid (except Gly) has a sidechain (or $R$ group) attached to its central atom $C_{\alpha}$. It is the sidechain and sidechain alone distinguishes one amino acid from another and provides a pseudo-atom (center of the sidechain). It has been argued in (Spriggs et al. 2003): “The vectorial representation is clearly an extremely simple description of the relative orientations of the side-chains in a 3D protein structure. It does, however, have the advantage that it does not overdefine the orientations of ends of side-chains, as should occur if a more precise representation was to be used that was based directly on the individual atomic coordinates in the PDB. This is a useful feature for at least three reasons: in medium-resolution protein-crystallographic studies, it is often difficult to get the final torsion-angle value correct and so the fixed orientations of the side-chain orientations may be in doubt; the identifications of individual atoms in a residue can often be ambiguous; and side-chains can often move or twist, for example, on binding substrates.”

There are currently over 30,000 proteins in the PDB database, containing 3D coordinates of all atoms in each protein. It is practical and relatively straightforward to build the vector model for each protein and calculate Euclidean distances between atoms. For the rest of this paper, a protein always means its vector model. We adopt this approach as a basis of our model, which is formulated in the next section.
3 Problem Formulation

This section presents a protein structure matching problem, which has been first introduced in (Huang et al. 2005). The problem essentially deals with the identification of matching structural regions, called “patches”, between two proteins.

3.1 Vector Representation of Protein Structures

A protein can be defined as a set $P$ of three dimensional vectors:

$$ P = \{ v_i | 1 \leq i \leq N \} \quad (1) $$

where $N = |P|$. Each $v_i$ denotes a vector of $C_{\alpha}C_{\beta}$ for amino acid $i$ (Fig.1(b)). The length of a vector (i.e., the distance between its $\alpha$-end and $\beta$-end) is typically fixed at 1.5 Å (angstrom).

3.2 Characterizing Protein Structures via patch Signatures

Since the proteins can be represented as geometric objects, the structures of the geometric objects have a direct influence on the proteins structure matching. We propose that 3D protein structure comparison can be performed by comparing the spatial relationship among vectors between two proteins. In other words, if two protein structures are similar, the spatial relationship among vectors of one structure must be similar to that of the other. The notion of characterization of spatial relationship refers to constraints which tie the vectors so that they have a fixed spatial relationship. That is, they can only rotate or translate globally as a whole without any internal change of positions. As the distances between atoms play a significant role in protein structure analysis, here we consider a distance-based characterization of spatial relationships between vectors. Since the PDB (Protein Data Bank) supplies coordinates of each atom of proteins in three-dimensional space, it is easy to calculate Euclidean distances between atoms.

The structural regions on a protein can be described as patches (Huang et al. 2005) which are subsets of vectors in the protein structure within a certain distance cutoff.

**Definition 1 (Patch).** A patch is defined as a spherical region of protein $P$, whose diameter is $\epsilon$ (i.e. a distance cut-off) (Fig.2(b)). More formally, $M = \{ v_1, v_2, ..., v_Q \} \subseteq P$ ($Q > 2$) is a patch if $(\forall v_i, v_j \in M, d_{\alpha\alpha}^{j,i} \leq \epsilon)$. In addition, $M$ is called a non-extendable patch if and only if $(\forall v_i, v_j \in M, d_{\alpha\alpha}^{j,i} \leq \epsilon)$ and $(\forall v_k \in M, \forall v_l \notin M, d_{\alpha\alpha}^{k,l} > \epsilon)$.

We can observe from the above definition that a non-extendable patch actually represents a maximal structural region with respect to a distance cutoff $\epsilon$ (15Å in this paper). Generally, a patch is a set of vectors with particular constraints on spatial arrangement.

**Proposition 1.** For a $k$-sized patch ($k > 2$), $7k - 10$ internal distances are sufficient to characterize the spatial relationship among the vectors.

A formal proof of this proposition can be found in (Huang 2005). As an example, we can look at one way of introducing the internal distances, as illustrated in Fig.3 (the dashed lines as internal distances). The first two vectors $v_1$ and $v_2$ form a stable triangular pyramid with the internal distances among their ends (Fig.3(a)). When the $i^{th}$ ($i > 2$) vector $v_i$ comes in, it constructs two triangular pyramids for tying to the original structure (i.e. $v_1$ and $v_2$), with four internal distances $d_{\alpha\alpha}^{1,i}, d_{\alpha\alpha}^{2,i}, d_{\beta\beta}^{1,i}, d_{\beta\beta}^{2,i}$ and other three distances $d_{\alpha\alpha}^{2,i}, d_{\beta\beta}^{2,i}$ and $d_{\beta\beta}^{1,i}$ (Fig.3(e)). Therefore, for $k$ vectors, the total number of internal distances is $4 + (k - 2) \times 4 + (k - 2) \times 3 = 7k - 10$.

**Proof.** Proof omitted. Refer to (Huang 2005) for details.

For a $k$-sized patch ($k > 2$), the set of $7k - 10$ internal distances is called its patch signature, which identifies the spatial relationship between vectors in the patch. Proposition 1 proves that $O(k)$ distances are required. It is of significance because the patch matching algorithms presented later are based on a number of internal distances linear to $k$. The fewer distances involved, the faster in patch comparison.

A $k$-sized patch is an unordered collection of $k$ vectors and in theory it has $k!$ representations of $7k - 10$ distances. Ordering the vectors is necessary for generating a unique representation of the patch. To generate an ordering of vectors in a patch, a base vector $v_1$ needs to be selected as a starting point, based on which an ordering function $q: q \rightarrow q'$ with $q'q = 1, k$ is defined. An detailed ordering algorithm was given in (Huang 2005). Throughout the rest of the paper, we assume that the vectors in any $k$-sized patch $S_{<}$ are already ordered, denoted as $S_{<}$. Now consider a $k$-sized patch $S_{\leq} = \{ v_1, v_2, ..., v_k \}$. According to Proposition 1, $S_{\leq}$ can be represented uniquely by $7k - 10$ distances (dimensions) in the following order:

$$ S_{\leq} = < d_{\alpha\alpha}^{1,2}, d_{\alpha\alpha}^{1,3}, ..., d_{\alpha\alpha}^{1,k}, d_{\alpha\alpha}^{2,3}, ..., d_{\alpha\alpha}^{2,k}, d_{\alpha\alpha}^{3,4}, ..., d_{\alpha\alpha}^{3,k}, d_{\beta\beta}^{1,2}, d_{\beta\beta}^{1,3}, ..., d_{\beta\beta}^{1,k}, d_{\beta\beta}^{2,3}, ..., d_{\beta\beta}^{2,k}, d_{\beta\beta}^{3,4}, ..., d_{\beta\beta}^{3,k} >. $$
Recall that in Section 2 we have mentioned the distance matrix approach, which would require three matrices to store all the $d_{αα}$, $d_{ββ}$, and $d_{αβ}$ distances. The advantage of our patch signature model lies in its linear representation, based on which we shall develop more efficient patch comparison algorithms.

3.3 Patch Matching

The following is the definition of a matching function between two patch signatures.

**Definition 2 (k-sized patch matching).** Given two $k$-sized patches, $S_{α} = \{v_{1}, v_{i}, ..., v_{k}\}$ and $S_{β} = \{u_{1}, u_{2}, ..., u_{k}\}$, both represented by their $7k$-dimensional patch signatures, i.e., $S_{α} = < s_{1}, s_{2}, ..., s_{n} >$ and $S_{β} = < s'_{1}, s'_{2}, ..., s'_{n} >$. They match each other (denoted as $S_{α} ≈ S_{β}$ or in short $S_{α} ≈ S_{β}$) if

$$s_{1} ≈ s'_{1} \land s_{2} ≈ s'_{2} \land ... s_{n} ≈ s'_{n} \quad (2)$$

where "≈" means "equals to within a tolerance $δ$".

Based on the k-sized patch matching, we can then define the non-extendable patch and protein structure matchings as follows.

**Definition 3 (Non-extendable Patch Matching).** For two non-extendable patches $M$ and $M'$, they match each other ($M ≈ M'$) if there exists a k-sized patch $S \subseteq M$ and another k-sized patch $S' \subseteq M'$ such that $S_{α} ≈ S_{β}$ and $5 \leq k \leq 20$.

**Definition 4 (Protein structure matching).** For two proteins $P$ and $P'$, they have a matching structure if there exists non-extendable patches $M \subseteq P$ and $M' \subseteq P'$ such that $M ≈ M'$.

In summary, given a query protein $Q$, the general problem we investigate is to find all the proteins from a protein database such that the resultant proteins have a one or more matching non-extendable patches with $Q$, and to identify all the maximum sized matching patches. The maximum sized matching patches are of interest and will be presented to the biologists for post-processing and further investigation.

3.4 A Match-and-Expand Strategy

In this subsection, we introduce a match-and-expand strategy for fast protein structure matching.

If two non-extendable patches $M$ and $M'$ have a maximal matching patch of $K$ vectors, they must also have matching sub-patches of $1, 2, ..., K - 1$ vectors. The match-and-expand strategy, similar to the philosophy of BLAST system (Altschul, Gish, Miller, Myers & Lipman 1990), first matches patches of the size $k (k \leq K)$ to reduce the number of candidates. A set of all patches of size $k$ is pre-computed for all proteins in the database. In order to check if $M$ and $M'$ have a matching patch, the $k$-sized patches of $M$ and $M'$ are checked first. If no $k$-sized matching sub-patches are found, $M$ and $M'$ will not have any matching sub-patches. Otherwise, $M$ and $M'$ will be further checked in the expand step, starting from their matching $k$-sized patches, until finding maximum $K$ sized matching patches. Operationally the expand stage can be accomplished by incrementally expanding $k$-sized sub-patches $S$ and $S'$ by one vector each time until maximum matching patches are reached.

The choice of $k$ is important. If it is too small, then the match step may generate too many false hits; if it is too large, then the cost of materializing all $k$-sized patches can be very high. However, the choice of $k$ is beyond the scope of this paper. We will focus on the match step.

We have defined the patch signature which is linear with respect to the number of atoms within a patch. The two $k$-sized patches can then be matched by comparing their patch signatures. Though the dimensionality of this representation ($7k - 10$) is much less than the traditional inter-atom distance matrix ($C_{2k}$), searching a large patch database is still expensive when $k$ is large. A obvious solution to the problem is to reduce the dimensionality of patch signatures while maximally preserving the matching function between two patches in the lower-dimensional space. We will study two powerful dimensionality reduction approaches in the next section and discuss how to apply them on patch signature data.

4 Dimensionality Reduction on Patch Signatures

Dimensionality reduction has been extensively applied in information retrieval. The goal is to find an “intrinsic” subspace, which is an approximation of the original space but with a lower dimensionality. It has been demonstrated that there does exist an intrinsic semantic sub-space where the dimensions with lower eigenvalues carry redundant information and therefore can be truncated (Ding 1999).

On the other hand, projecting the original data to a lower dimensional space also helps discover some embedded “latent semantics” - i.e., some implicit associations which are unseen in the original high dimensional space.

A well known dimensionality reduction approach is the Singular Value Decomposition (SVD), which is the basis of the Latent Semantic Indexing (LSI) (Deerwester et al. 1990) (Landauer et al. 1998).

Recently, a Locality Preserving Projection (LPP) algorithm (He et al. 2004) has been introduced for document indexing and demonstrated better performance than SVD. Unlike SVD, which preserves inner product in an Euclidean space, the LPP aims to preserve local geometrical structure of data manifold.

Note that our patch signature matching function, defined in the last section, requires that the difference between values of each dimension of two data points should be within a tolerance. Neither SVD nor LPP is designed to directly preserve such a matching function. Therefore, we propose to use the Euclidean distance based measure between two patch signatures as an approximation of the previous pairwise matching function. Since $k$-sized patches can be equivalently treated as points in a $7k - 10$ dimensional space, the similarity between two patches can then be measured by the Euclidean distance between them.

**Definition 5 (Patch Similarity $\sim_{δ}$.** Given two $k$-sized patches $S_{α} = < s_{1}, s_{2}, ..., s_{n} >$ and $S'_{α} = < s'_{1}, s'_{2}, ..., s'_{n} >$. They are similar (denoted as $S_{α} \sim_{δ} S'_{α}$ or in short $S_{α} \sim S'_{α}$) if $d_{2}(S_{α}, S'_{α}) < δ'$, where

$$d_{2}(S_{α}, S'_{α}) = \sqrt{\sum_{i=1}^{n} |s_{i} - s'_{i}|^2} \quad (3)$$
traces respectively and \( \Sigma \) is an \( \sqrt{n} \) of the singular values of whose values are monotonically increasing non-zero also adopt the experimental way.

The next two subsections will describe SVD and LPP algorithms respectively and give details in how they can be applied to the patch signature data.

### 4.1 Singular Value Decomposition (SVD)

Singular value decomposition (SVD) is a powerful technique from linear algebra. Given \( m \times n \) patch signature matrix \( X \) with rank \( r \), where \( m \) is the number of \( k \)-sized patches and \( n \) is the number (i.e., \( 7k-10 \)) of dimensions, \( X \) can be decomposed to:

\[
X = U \Sigma V^T
\]

where \( U \) and \( V \) are orthogonal \( m \times r \) and \( n \times r \) matrices respectively and \( \Sigma \) is an \( r \times r \) diagonal matrix whose values are monotonically increasing non-zero singular values of \( X \). The columns of \( U \) and \( V \) are the eigenvectors of \( XX^T \) and \( X^TX \) respectively.

Dimensional reduction is performed by taking only the first \( p \) eigenvectors and singular values to form:

\[
X_p = U_p \Sigma_p V_p^T
\]

where \( U_p \) and \( V_p \) are \( m \times p \) and \( n \times p \) matrices composed of the first \( p \) columns of \( U \) and \( V \) respectively. According to the Eckart-Young theorem, \( X_p \) is the closest rank-\( p \) approximation by least square method to \( X \) in sense of both matrix Frobenius norm and 2-norm, i.e.

\[
X_p = \min_{\text{rank}(B)=p} \| X - B \|_2
\]

\[
X_p = \min_{\text{rank}(B)=p} \| X - B \|_F
\]

Via SVD, the \( j \)-th patch signature vector \( S_{c,j} \) can be projected to a \( p \)-dimensional vector on the feature space of span \( V_p^T \). The projected vector is actually recorded as the \( j \)-th row of \( U_p \).

For a general exposition of the theory of SVD the reader is directed to (Golub & Van Loan 1996). The major difficulty of LSA is the choice of a suitable value for \( p \). Tough the choice of optimal \( p \) can be theoretical, for example the work by Ding (Ding 1999), experimental approach is more widely used in information retrieval community, where an optimal \( p \) is derived by reference to some experiment. In our experiments we also adopt the experimental way.

### 4.2 Locality Preserving Projection

Locality Preserving Projection (LPP) (He & Niyogi 2003)(He et al. 2004) aims to preserve the intrinsic geometric structure in term of local neighborhood information of the data on a manifold.

Suppose a set of \( n \)-dimensional patches \( x_1, x_2, \ldots, x_m \) in space \( \mathbb{R}^n \) form a \( m \times n \) data matrix \( X \). The core LPP algorithm includes the following steps:

1. Construct an adjacency graph with each data point (i.e., patch) as a node and put an edge between two point \( x_i \) and \( x_j \) if they are close enough. The closeness between \( x_i \) and \( x_j \) can be measured by their distance \( ||x_i - x_j||_2 \). A simple but effective way of connecting two nodes is based on \( q \) nearest neighbors, i.e., \( x_i \) and \( x_j \) are the \( q \) nearest neighboring points;
2. A \( m \times m \) adjacency matrix \( W \) is built whereby \( W(i,j) = 1 \) if \( x_i \) and \( x_j \) are connected; otherwise \( W(i,j) = 0 \).

There are some other options to the adjacency graph construction and adjacency matrix weighting. We do not compare these different options in this paper and will leave it as one of our future work.

3. Compute Eigenmaps by solving the following generalized eigenvector problem:

\[
XLX^T a_l = \lambda_l XDX^T a_l
\]

where \( D \) is \( m \times m \) diagonal matrix with \( D_{ij} = \sum_j W_{ji} \), \( L = D - W \) is the Laplacian matrix, \( \lambda_l \) is the \( l \)-th eigenvalue and \( a_l \) is the \( l \)-th eigenvector. The transformation matrix \( A = [a_1, a_2, \ldots, a_p] \) can be formed, ordered by the eigenvalues \( \lambda_1 < \lambda_2 < \ldots < \lambda_p \) where \( p << n \).

4. Project the points to \( p \)-dimensional space \( \mathbb{R}^p \):

\[
x_i \rightarrow x_i' = A^T x_i
\]

Note that LPP is a linear approximation of Laplacian Eigenmaps (Belkin & Niyogi 2001). They both try to preserve locality via the following objective function:

\[
\min_{ij} \| x_i' - x_j' \|^2 W_{ij}
\]

They are the same in the first two steps. The step 3 of the latter is to compute the generalized eigenvector for:

\[
La_l = \lambda_l Da_l
\]

The rows of resultant \( m \times p \) matrix \( A \) can be used as approximation of the original data in the lower dimensional space \( \mathbb{R}^p \):

\[
x_i' = (a_1(i), a_2(i), \ldots, a_p(i))
\]

The justification for their ability of preserving geometric structure on manifold is based on the Laplacian matrix \( L \) which is an approximation to the Laplace-Beltrami operator defined on the manifold (Belkin & Niyogi 2001).

### 5 Experiments

In this section, we set up the experiments and report the results of an extensive performance study conducted to evaluate the proposed representation model and the dimensionality reduction on protein patch data.

#### 5.1 Experimental setup

**5.1.1 Test Data**

A total number of 811 sample proteins are selected for our initial experiments according to the PDB_LIST_20040601 (R-factor<0.2 and...
There are several parameters need to be set for our model and search method, four of which are fixed in our experiments:

- Distance cutoff ($\varsigma$): 15Å
- Pair-wise matching tolerance ($\delta$): 4Å
- Size of patches to match ($k$): 16 (leading to a total dimensionality 102 for patch signatures)
- Number of nearest neighbors in LPP ($q$): 10

Two other parameters are variables. We will test how the different settings of them affect the performance.

- Euclidean distance based similarity threshold $\delta'$:
  - Size of reduced dimensionality $p$: 5, 10, 20, 30, 40, 50, 102

5.2 Experimental results

Table 2, 3, Fig.4, and Fig.5 summarize the experimental results. In addition, the CPU time for the baseline is 3.1 seconds. We can make the following observations:

Dimensionality reduction by both SVD and LPP under all the different parameter settings saves CPU time by from 3.2% up to 84%. This suggests that it largely improve the efficiency for patch matching.

Larger threshold value $\delta'$ lead to increasing CPU time and recall, and decreasing precision. This co-relates our intuition. According to proposition 3, a threshold $\delta' = \sqrt{n\delta} = 40Å$ guarantees 100% recall. The cost is losing precision. In the rest of our analysis, we take the F-measure as the main effectiveness indicator, as it represents the trade-off between precision and recall. We can observe that a much lower threshold like 1.2Å is enough to obtain reasonable F-value.

The “intrinsic” dimensionality for either LPP or SVD is quite low (20 for SVD and 10-20 for LPP). In Fig.5, for each model the F-value grows rapidly until it reaches the peak, where the corresponding dimensionality is the intrinsic dimensionality. After this certain point, the F-value decreases while the dimensionality increases. This suggests that a large number of less significant dimensions carry no much meaningful information. This also indicates the usefulness and necessity of dimensionality reduction. It is also interesting to note the difference between LPP and SVD.

The performance of SVD decreases more rapidly than LPP when the dimensionality increases. More theoretical comparison between the two approaches will be conducted in the future work.

6 Related Work

This paper deals with the problem of finding similar substructures. The most related techniques to our methods include protein structure modelling, such as geometric hashing and graph theoretical approach, and high-dimensional indexing for similarity search.

Geometric hashing (Wolfson 1997) was originally developed in computer vision and now used in protein structure comparison. It defines a set of reference frames for a structure. The coordinates of all points in the structure are re-calculated in a reference frame, forming a reference frame system. Geometric features of the structure are calculated based on the reference frame systems and stored in a hash table. This method ignores the sequential order of amino acids and gives the result invariant to the translation and rotation of the compared structures(Nussinov & Wolfson 1991) and thus is useful to discover matching substructures. However, we do not adopt this approach as it is computationally expensive. The number of reference frame systems to be constructed and the number of frame system comparisons are both combinatorial. Moreover, recalculation of the coordinates of points in a new reference frame via rotation and translation is also expensive.

The graph theoretical approach is used in systems, such as ASSAM(Spriggs et al. 2003, Grindley, Artyymiuk, Rice & Willett 1993) and VAST(Gibrat...
Table 2: Summary of SVD performance

<table>
<thead>
<tr>
<th>p</th>
<th>$\delta'$</th>
<th>CPU Time % of baseline CPU time</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>0.413%</td>
<td>0.21</td>
<td>0.79</td>
<td>0.33</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>0.413%</td>
<td>0.17</td>
<td>0.90</td>
<td>0.29</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>0.912%</td>
<td>0.05</td>
<td>0.99</td>
<td>0.10</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0.516%</td>
<td>0.7</td>
<td>0.23</td>
<td>0.30</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>0.516%</td>
<td>0.16</td>
<td>0.75</td>
<td>0.26</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>0.826%</td>
<td>0.04</td>
<td>0.99</td>
<td>0.08</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>0.512%</td>
<td>0.92</td>
<td>0.42</td>
<td>0.21</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>0.518%</td>
<td>0.28</td>
<td>0.46</td>
<td>0.35</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>1.280%</td>
<td>0.01</td>
<td>0.99</td>
<td>0.02</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>0.619%</td>
<td>0.85</td>
<td>0.39</td>
<td>0.07</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>0.619%</td>
<td>0.24</td>
<td>0.26</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>1.32%</td>
<td>0.02</td>
<td>0.99</td>
<td>0.04</td>
</tr>
<tr>
<td>40</td>
<td>1</td>
<td>0.826%</td>
<td>0.22</td>
<td>0.003</td>
<td>0.06</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>0.826%</td>
<td>0.31</td>
<td>0.12</td>
<td>0.17</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>2.64%</td>
<td>0.02</td>
<td>0.98</td>
<td>0.04</td>
</tr>
<tr>
<td>50</td>
<td>1</td>
<td>0.826%</td>
<td>0.3</td>
<td>0.002</td>
<td>0.004</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>0.929%</td>
<td>0.23</td>
<td>0.05</td>
<td>0.08</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>1.548%</td>
<td>0.03</td>
<td>0.98</td>
<td>0.06</td>
</tr>
<tr>
<td>102</td>
<td>1</td>
<td>1.135%</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>1.342%</td>
<td>0.14</td>
<td>0.001</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>2.581%</td>
<td>0.08</td>
<td>0.98</td>
<td>0.15</td>
</tr>
</tbody>
</table>

Table 3: Summary of LPP performance

<table>
<thead>
<tr>
<th>p</th>
<th>$\delta'$</th>
<th>CPU Time % of baseline CPU time</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>0.516%</td>
<td>0.14</td>
<td>0.74</td>
<td>0.24</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>0.516%</td>
<td>0.07</td>
<td>0.77</td>
<td>0.13</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>0.619%</td>
<td>0.02</td>
<td>0.96</td>
<td>0.04</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0.516%</td>
<td>0.25</td>
<td>0.47</td>
<td>0.33</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>0.516%</td>
<td>0.20</td>
<td>0.52</td>
<td>0.29</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>0.826%</td>
<td>0.01</td>
<td>0.96</td>
<td>0.02</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>0.826%</td>
<td>0.37</td>
<td>0.25</td>
<td>0.30</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>0.826%</td>
<td>0.32</td>
<td>0.54</td>
<td>0.33</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>1.32%</td>
<td>0.02</td>
<td>0.94</td>
<td>0.04</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>0.619%</td>
<td>0.51</td>
<td>0.16</td>
<td>0.24</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>0.723%</td>
<td>0.43</td>
<td>0.21</td>
<td>0.28</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>2.168%</td>
<td>0.02</td>
<td>0.97</td>
<td>0.04</td>
</tr>
<tr>
<td>40</td>
<td>1</td>
<td>0.723%</td>
<td>0.62</td>
<td>0.11</td>
<td>0.19</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>0.723%</td>
<td>0.57</td>
<td>0.18</td>
<td>0.27</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>1.342%</td>
<td>0.02</td>
<td>0.98</td>
<td>0.04</td>
</tr>
<tr>
<td>50</td>
<td>1</td>
<td>0.723%</td>
<td>0.67</td>
<td>0.09</td>
<td>0.16</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>1.32%</td>
<td>0.53</td>
<td>0.15</td>
<td>0.23</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>1.335%</td>
<td>0.02</td>
<td>0.96</td>
<td>0.04</td>
</tr>
<tr>
<td>102</td>
<td>1</td>
<td>1.32%</td>
<td>0.78</td>
<td>0.05</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>1.32%</td>
<td>0.65</td>
<td>0.09</td>
<td>0.16</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>2.168%</td>
<td>0.02</td>
<td>0.96</td>
<td>0.04</td>
</tr>
</tbody>
</table>

(a) SVD performance  (b) LPP performance

Figure 4: F-Measure.
et al. 1996), to find the maximal common substructure. The problem is often transformed to the clique problem. Vector representation is also used in ASSAM (Spriggs et al. 2003) which is developed to search for patterns of amino acid side chains in the 3D structures. A substructure is characterized by the distances among all pairs of vectors. This is therefore different from our model, where the overall spatial relationship of all the vectors in the substructure is characterized by its patch signature to make a more compact representation. ASSAM then detects cliques using a maximal common subgraph isomorphism algorithm borrowed from graph theory (Bron & Kerbosch 1973). As the clique detection problem is NP-Complete, many heuristic algorithms are developed. The most existing heuristic algorithms for the clique problem are partially enumerative and branch-and-bound based (Gardiner, Artymiuk & Willett 1997). However, they are insufficient to handle large scale data. For example, the test queries used for the experiments reported in (Spriggs et al. 2003) were all triad residues. In other words, the maximum size of cliques was three. A protein was “hit” once a matching substructure of size 3 was found. In our work, a query is a whole protein and we aim to find from the database all the matching substructures in any size. Therefore, we do not use the clique detection algorithms in our work. Instead, we developed a more scalable IR and database solution featured by a highly efficient query processing strategy.

7 Conclusions and Future Work

This paper presents a protein structure matching problem and formulates it as an information retrieval problem. A patch signature model is addressed based on a vector representation of protein structure. A protein structural region is defined as a patch, formed by a set of vectors within the region. A k-sized patch is then indexed by the $7k − 10$ internal inter-atom distances constituting its patch signature. A matching function is defined to compare two patches based on their patch signatures. Though the dimensionality of this representation ($7k − 10$) is much less than the traditional inter-atom distance matrix ($C_{n^2}$) approach, searching a large patch database is still expensive when $k$ is large. We propose to apply dimensionality reduction to patch signatures and show how the two problems are adapted to fit each other. The Locality Preservation Projection (LPP) and Singular Value Decomposition (SVD) are chosen and tested for this purpose. Experimental results show that the dimensionality reduction improves the searching speed with acceptable precision and recall. From a more general point of view, this paper demonstrates that information retrieval techniques can play a crucial role in solving this biologically critical but previously computationally prohibitive problem. It is our hope that the marriage between information retrieval and bio-informatics will extend the boundaries of both areas.

From the experimental results, we can observe that there is still some room for further performance improvement in dimensionality reduction via both LPP and SVD (The best F-values are separately 33% and 35%). We will investigate other possibly more effective approximations to the pairwise patch matching function, other than the Euclidean distance used in this paper. On the other hand, more dimensionality reduction algorithms will be studied. At this stage, we focus on matching same sized patches. In the future, we plan to develop an efficient indexing mechanism for different sized patches. In this paper, we did not compare our approach to other protein structure matching algorithms. As a future work, we will also consider testing our approach on a collection of “homologs” produced from the SCOP database.

References


Figure 5: Intrinsic dimensionality ($\delta' = 1.2$).


