Dimensionality Reduction in Patch-Signature Based Protein Structure Matching

Zi Huang[1] **Xiaofang Zhou**[1] **Dawei Song**[2] **Peter Bruza**[2]

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

[2]Distributed 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 similarly 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 characterizes 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-

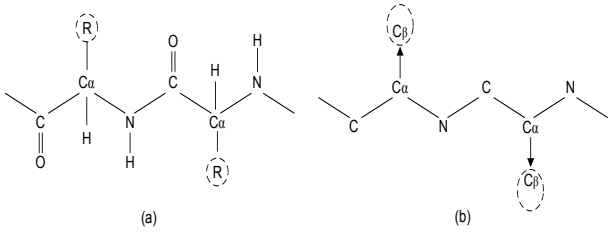


Figure 1: A fragment of amino acid chain.

tion (SVD), which best preserves inner product in an Euclidean space and is the basis of the Latent Semantic Indexing (LSI)(Deerwester, Dumais, Landauer, Furnas & Harshman 1990)(Landauer, Foltz & Laham 1998). Recently, a novel Locality Preserving Projection (LPP) algorithm (He, Cai, Liu & Ma 2004) has been introduced to document indexing and demonstrated a better performance. Unlike SVD, LPP aims to preserve local geometrical structure in a manifold in terms of L_2 distance between data points. In this paper, we will address how the LPP and SVD can be applied to patch matching and demonstrate that they can largely improve efficiency (measured by CPU time) while maintaining an acceptable precision and recall.

The rest of the paper is organized as follows. Section 2 gives a brief introduction to protein structure and its 3D representations. We present a patch signature model and its similarity measure in Section 3. Two typical dimensionality reduction approaches, SVD and LPP, and their application in patch signature matching are studied in Section 4. Section 5 reports the experimental results. The related work are described in section 6. Section 7 finally concludes the paper and highlights the future work.

2 Preliminaries

A protein is a large molecule composed of one or more chains of amino acids in a specific order. Twenty standard amino acids have been identified in protein structures. As illustrated in Fig. 1(a), each amino acid contains a central atom C_α to which an amino (N-H) group and a Carboxyl ($C=O$) group are attached. The amino group, carboxyl group and C_α atom construct the *mainchain*(or backbone) of an amino acid. In addition, each amino acid (except Gly) has a sidechain (or R group) attached to its central atom C_α. It is the sidechain and sidechain alone which distinguishes one amino acid from another and furthermore confers the specific function to an amino acid(Bourne & Weissig 2003). The *sidechain* is typically connected to C_α via another atom C_β(Branden & Tooze 1998). A protein is constructed by amino acids that are linked by peptide bonds forming a polypeptide chain.

The amino acid sequence of a protein's polypeptide chain is called its primary structure, which can be represented a linear string of amino acids, abbreviated with one-letter codes.

Protein structure can be folded into a three-dimensional configuration as a set of points (atoms) in 3D space. For example, PDB (Protein Data Bank)(*Protein data Bank* n.d.) arranges a protein on an imaginary Cartesian coordinate frame and assigns (x,y,z) coordinates to each atom. This representation serves as a basis of different higher level representations. Different regions on the amino acid sequence form regular secondary structures, including the α helices and β sheets in the three-dimensional space. A 3D protein structure can usually be characterized by its mainchain (via C_α atoms) and/or sidechains (via C_β atoms).

For example, in the DALI(Holm & Sander 1993)(Holm & Sander 1996) system, a distance matrix containing all pairwise distances between C_α atoms is built, where each C_α-C_α distance reflects the relationship of two amino acids respectively centered by the two C_α atoms. If the distance between two amino acids (A_i and A_j) of protein A is similar to the distance between two amino acids (B_i and B_j) of protein B, amino acids A_i and A_j could be mapped to the amino acids B_i and B_j.

The VAST(Gibrat, Madej & Bryant 1996) and SARF(Alexandrov & Fischer 1996) systems use secondary structural elements (SSE). Each SSE in a protein is represented by position, length, and direction of a vector determined by the position of the C_α atoms along the SSE. It assumes that if two vectors representing two secondary structures are similar, the internal structure within secondary structures are similar.

The program SSAP(Orengo & Taylor 1996)(Orengo & Taylor 1989) represents 3D structure of protein as structural environments for amino acids, each of which is the set of vectors from the C_β atom to C_β atoms of all other amino acids in the protein.

There are some other methods such as Torsion (dihedral) Angles (Bergeron 2003). However, all the above methods are based on either mainchains (via C_α) or sidechains (via C_β) alone, thus they are insufficient to model the orientation of sidechains. A different way of representing a protein's structure as vectors of C_α-C_β atoms. A pair of C_α-C_β atoms in the same amino acid constructs a vector, denoted $\overrightarrow{C_\alpha C_\beta}$, from its C_α end to C_β end. More recently, the vector representation model(Spriggs, Artymiuk & Willett 2003, Huang, Zhou & Song 2005) has been operationalized. For each residue, a vector from C_α to C_β can be constructed. This vector representation involves not only the mainchain but also the sidechain information. The position of C_β atom is used to emphasize the functional part of the side-chain corresponding to the vector. The vector representation also offers a flexibility of generalizing the use of C_β to a psudo-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 chould 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 oftern difficult to get the final torsion-angle value correct and so the fine details of the side-chain orientations may be in doubt; the identifications of individual atoms in a residue can oftern be ambiguous; and side-chains can often move or twist, for exampe, 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.

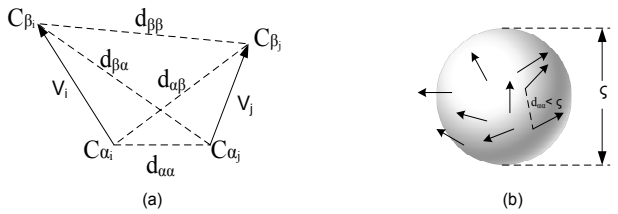


Figure 2: (a)Spatial relationship between two vectors. Four internal distances are denoted as $d_{\alpha\alpha}$, $d_{\beta\beta}$, $d_{\alpha\beta}$, and $d_{\beta\alpha}$. (b) An example of patch. Each vector represents an amino acid. The diameter is ε. The dashed line shows the α-α distance ($d_{\alpha\alpha}$) between two vectors.

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 $\overrightarrow{C_\alpha C_\beta}$ for amino acid i (Fig.1(b)). The length of a vector (i.e., the distance between its α-end and β-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 ε (ie. a distance cut-off) (Fig.2)(b). More formally, $M = \{v_1, v_2, ..., v_Q\} \subseteq P$ $(Q > 2)$ is a patch if $(\forall v_i, v_j \in M, d^{i,j}_{\alpha\alpha} \leq \varsigma)$. In addition, M is called a non-extendable patch if and only if $(\forall v_i, v_j \in M, d^{i,j}_{\alpha\alpha} \leq \varsigma) \wedge (\forall v_k \in M, \forall v_l \notin M, d^{k,l}_{\alpha\alpha} > \varsigma)$.*

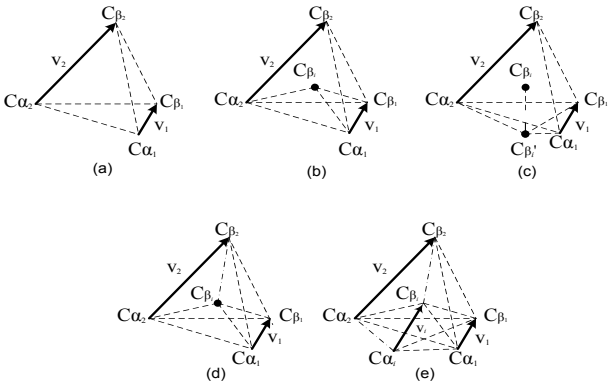


Figure 3: Patch Signature.

We can observe from the above definition that a non-extendable patch actually represents a maximal structural region with respect to a distance cutoff ς ($15\mathring{A}$ 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 $d^{v_1,v_i}_{\alpha\alpha}$, $d^{v_1,v_i}_{\beta\beta}$, $d^{v_1,v_i}_{\alpha\beta}$, $d^{v_1,v_i}_{\beta\alpha}$, and other three distances $d^{v_2,v_i}_{\alpha\alpha}$, $d^{v_2,v_i}_{\beta\beta}$ and $d^{v_2,v_i}_{\alpha\beta}$ (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 *basevector* v_{i_b} needs to be selected as a starting point, based on which an ordering function $\phi_{i_b} : q \rightarrow q' | q, q' = 1..k$ 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 $S_\triangleleft$. Now consider a k-sized patch $S_\triangleleft = \{v_{i_1}, v_{i_2}, ..., v_{i_k}\}$. According to Proposition 1, $S_\triangleleft$ can be represented uniquely by $7k - 10$ distances (dimensions) in the following order:

$$S_\triangleleft = < d^{i_1,i_2}_{\alpha\alpha}, d^{i_1,i_3}_{\alpha\alpha}, ..., d^{i_1,i_k}_{\alpha\alpha}, d^{i_2,i_3}_{\alpha\alpha}, ..., d^{i_2,i_k}_{\alpha\alpha}, d^{i_1,i_2}_{\beta\beta}, d^{i_1,i_3}_{\beta\beta}, ..., d^{i_1,i_k}_{\beta\beta}, d^{i_2,i_3}_{\beta\beta}, ..., d^{i_2,i_k}_{\beta\beta},$$

$d_{\alpha\beta}^{i_1,i_2}, d_{\alpha\beta}^{i_1,i_3}, ..., d_{\alpha\beta}^{i_1,i_k}$,
$d_{\beta\alpha}^{i_1,i_2}, d_{\beta\alpha}^{i_1,i_3}, ..., d_{\beta\alpha}^{i_1,i_k} >$

Recall that in Section 2 we have mentioned the distance matrix approach, which would require three matrices to store all the $d_{\alpha\alpha}$, $d_{\beta\beta}$, $d_{\alpha\beta}$, and $C_{\beta\alpha}$ 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, $S_\triangleleft = \{v_{i_1}, v_{i_2}, ..., v_{i_k}\}$ and $S'_\triangleleft = \{u_{i_1}, u_{i_2}, ..., u_{i_k}\}$, both represented by their $7k - 10$ dimensional patch signatures, i.e. $S_\triangleleft =< s_1, s_2, ..., s_n >$ and $S'_\triangleleft =< s'_1, s'_2, ..., s'_n >$. They match each other (denoted as $S_\triangleleft \approx_\delta S'_\triangleleft$ or in short $S_\triangleleft \approx S'_\triangleleft$) if*

$$s_1 \approx s'_1 \wedge s_2 \approx s'_2 \wedge ...s_n \approx s'_n \quad (2)$$

where "$\approx$" 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 ($M \approx_\delta M'$) if there exists a k-sized patch $S \subseteq M$ and another k-sized patch $S' \subseteq M'$ such that $S_\triangleleft \approx_\delta S'_\triangleleft$ and $5 < k \leq 20$.*

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 $M \approx_\delta M'$.*

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, $\cdots$, $K-1$ 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}^2), 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 $\sim_{\delta'}$)**.** *Given two k-sized patches $S_\triangleleft =< s_1, s_2, ..., s_n >$ and $S'_\triangleleft =< s'_1, s'_2, ..., s'_n >$. They are similar (denoted as $S_\triangleleft \sim_{\delta'} S'_\triangleleft$ or in short $S_\triangleleft \sim S'_\triangleleft$) if $d_2(S_\triangleleft, S'_\triangleleft) < \delta'$, where*

$$d_2(S_\triangleleft, S'_\triangleleft) = \sqrt{\sum_{i=1}^{n} |s_i - s'_i|^2} \quad (3)$$

Next, we will show theoretically how the Euclidean distance based similarity measure can return a super-set of the resultant matches from the pairwise matching and thus guarantees the recall of matching results.

Proposition 2. *If* $S_{\lhd} \approx_{\delta} S'_{\lhd}$, *then* $d_2(S_{\lhd}, S'_{\lhd}) < \sqrt{n}\delta$

Proof. This proposition can be proven trivially according to definition 3 and definition 6. □

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:

$$X = U\Sigma V^T \tag{4}$$

where U and V are orthogonal $m \times r$ and $n \times r$ matrices respectively and Σ 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 eigen vectors and singular values to form:

$$X_p = U_p\Sigma_p V_p^T \tag{5}$$

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.

$$X_p = \min_{rank(B)=p} ||X - B||_2 \tag{6}$$

$$X_p = \min_{rank(B)=p} ||X - B||_F \tag{7}$$

Via SVD, the $j-th$ patch signature vector $S_{\lhd j}$ 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 $x_1, x_2, ..., x_m$ in space $\Re^n$ 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:

$$XLX^T a_l = \lambda_l XDX^T a_l \tag{8}$$

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

4. Project the points to p-dimensional space $\Re^p$:

$$x_i \rightarrow x'_i = A^T x_i \tag{9}$$

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

$$min \sum_{ij} (x'_i - x'_j)^2 W_{ij} \tag{10}$$

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

$$La_l = \lambda_l Da_l \tag{11}$$

The rows of resultant $m \times p$ matrix A can be used as approximation of the original data in the lower dimensional space $\Re^p$:

$$x'_i = (a_1(i), a_2(i), ..., a_p(i)) \tag{12}$$

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

Table 1: Statistics of test data

Total number of proteins	811
Total number of vectors	190,669
Average number of vectors per protein	216
Average number of 16-sized patches per protein	5308

Resolution<1.9) in the WHATIF relational database. The PDB structures stored in the WHAT IF relational database are a representative set of sequence-unique (a sequence identity percentage cutoff of 30%) structures(*WHATIF relational database* n.d.). After pre-processing, the data statistics are shown in Table 1.

5.1.2 Query proteins

Ten different sized proteins are selected as queries. The average number of vectors per query is 238.

5.1.3 Baseline

To choose a baseline for comparison with our method, we perform pairwise matching of all distances between two patch signatures. The baseline matching results are assumed "correct matches".

The models we test in our experiments are the Euclidean distance based similarity search based on

- Dimensionality Reduction via SVD

- Dimensionality Reduction via LPP

5.1.4 Performance Indicators

Our programs are written in C++ and running on Pentium 4 CPU (2.8GHZ) with 1G RAM. The major performance indicators we used are:

- CPU time (in seconds) to complete a query

- Precision: percentage of returned patches being correct

- Recall: percentage of correct matching patches being returned

- F-measure: $\frac{2*Precision*Recall}{Precision+Recall}$

Note that all the experimental results reported later will be averaged for one query protein matches against one protein in the database.

5.1.5 Parameter settings

There are several parameters need to be set for our model and search method, four of which are fixed in our experiments:

- Distance cutoff (ς): $15\mathring{A}$

- Pair-wise matching tolerance (δ): $4\mathring{A}$

- 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 δ': $1\mathring{A}$, $1.2\mathring{A}$, $1.5\mathring{A}$

- 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 does largely improve the efficiency for patch matching.

Larger threshold value δ' lead to increasing CPU time and recall, and decreasing precision. This co-relates our intuition. According to proposition 3, a threshold $\delta' = \sqrt{n}\delta = 40\mathring{A}$ 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 $1.2\mathring{A}$ 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.

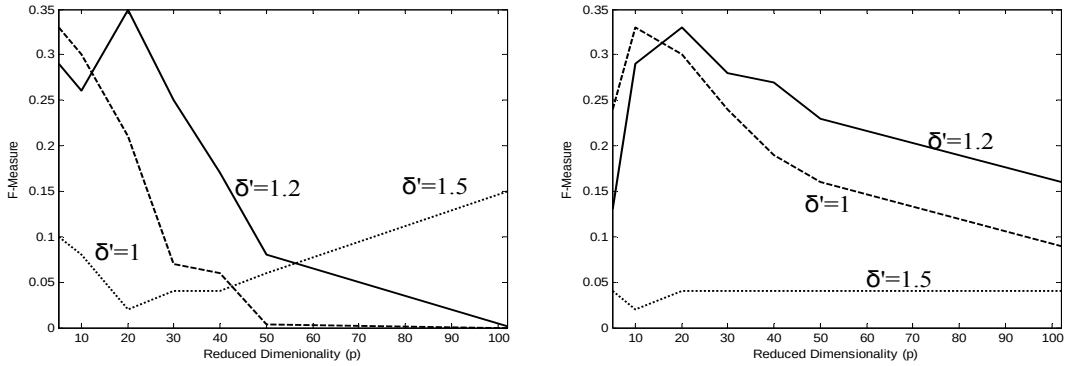
The graph theoretical approach is used in systems, such as ASSAM(Spriggs et al. 2003, Grindley, Artymiuk, Rice & Willett 1993) and VAST(Gibrat

Table 2: Summary of SVD performance

p	δ'	CPU Time $\%ofbaselineCPUtime$	Precision	Recall	F-Measure
5	1	0.4 13%	0.21	0.79	0.33
	1.2	0.4 13%	0.17	0.90	0.29
	1.5	0.9 29%	0.05	0.99	0.10
10	1	0.5 16%	0.7	0.23	0.30
	1.2	0.5 16%	0.16	0.75	0.26
	1.5	0.8 26%	0.04	0.99	0.08
20	1	0.52 17%	0.92	0.42	0.21
	1.2	0.57 18%	0.28	0.46	0.35
	1.5	1.2 80%	0.01	0.99	0.02
30	1	0.6 19%	0.85	0.39	0.07
	1.2	0.6 19%	0.24	0.26	0.25
	1.5	1 32%	0.02	0.99	0.04
40	1	0.8 26%	0.22	0.003	0.06
	1.2	0.8 26%	0.31	0.12	0.17
	1.5	2 64%	0.02	0.98	0.04
50	1	0.8 26%	0.3	0.002	0.004
	1.2	0.9 29%	0.23	0.05	0.08
	1.5	1.5 48%	0.03	0.98	0.06
102	1	1.1 35%	0	0	N/A
	1.2	1.3 42%	0.14	0.001	0.002
	1.5	2.5 81%	0.08	0.98	0.15

Table 3: Summary of LPP performance

p	δ'	CPU Time $\%ofbaselineCPUtime$	Precision	Recall	F-Measure
5	1	0.5 16%	0.14	0.74	0.24
	1.2	0.5 16%	0.07	0.77	0.13
	1.5	0.6 19%	0.02	0.96	0.04
10	1	0.5 16%	0.25	0.47	0.33
	1.2	0.5 16%	0.20	0.52	0.29
	1.5	0.8 26%	0.01	0.96	0.02
20	1	0.8 26%	0.37	0.25	0.30
	1.2	0.8 26%	0.32	0.34	0.33
	1.5	1 32%	0.02	0.94	0.04
30	1	0.6 19%	0.51	0.16	0.24
	1.2	0.7 23%	0.43	0.21	0.28
	1.5	2.1 68%	0.02	0.97	0.04
40	1	0.7 23%	0.62	0.11	0.19
	1.2	0.7 23%	0.57	0.18	0.27
	1.5	1.3 42%	0.02	0.98	0.04
50	1	0.7 23%	0.67	0.09	0.16
	1.2	1 32%	0.53	0.15	0.23
	1.5	1.1 35%	0.02	0.96	0.04
102	1	1 32%	0.78	0.05	0.09
	1.2	1 32%	0.65	0.09	0.16
	1.5	2.1 68%	0.02	0.96	0.04



(a) SVD performance (b) LPP performance

Figure 4: F-Measure.

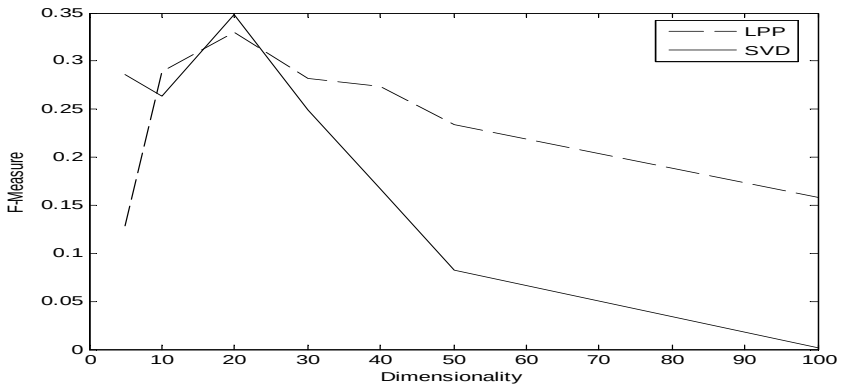


Figure 5: Intrinsic dimensionality(δ'=1.2).

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_{2k}^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

Alexandrov, N. & Fischer, D. (1996), 'Analysis of topological and nontopological structural similarities in the pdb: New examples with old structures', *Proteins* **25**, 354–365.

Altschul, S., Gish, W., Miller, W., Myers, E. & Lipman, D. (1990), 'Basic local alignment search tool', *J Mol Biol* **215**(3), 403–10.

Belkin, M. & Niyogi, P. (2001), Laplacian eigenmaps and spectral techniques for embedding and clustering, *in* 'Advances in Neutral Information Processing Systems 14 $NIPS$2001'.

Bergeron, B. (2003), *Bioinformatics Computing*, Pearson Education, Inc.

Bourne, P. & Weissig, H. (2003), *Structural Bioinformatics*, John Wiley and Sons.

Branden, C. & Tooze, J. (1998), *Introduction to Protein Structure*, Garland Publishing, Inc.

Bron, C. & Kerbosch, J. (1973), 'Algorithm 457 - finding all cliques of an undirected graph', *Communications of ACM* **1973**(16), 575–577.

Deerwester, S., Dumais, S., Landauer, T., Furnas, G. & Harshman, R. (1990), 'Indexing by latent semantic analysis', *Journal of American Society for Information Science* **41**, 391–407.

Ding, C. (1999), A similarity-based probability model for latent semantic indexing, *in* 'Proceedings of the Tweenty-Second Annual Internation ACM SIGIR Conference on Research and Development in Information Retrieval', pp. 59–65.

Gardiner, E., Artymiuk, P. & Willett, P. (1997), 'Clique-dection algorithms for matching three-dimensional molecular structures.', *Journal of Molecular Graphics and Modeling* **15**, 245–253.

Gibrat, J.-F., Madej, T. & Bryant, S. (1996), 'Surprising similarities in structure comparison', *Curr. Opin. Struct. Biol.* **6**, 377–385.

Golub, G. & Van Loan, C. (1996), *Matrix Computations*, John Hopkins University Press.

Grindley, H., Artymiuk, P., Rice, D. & Willett, P. (1993), 'Use of techniques derived from graph theory to compare secondary structure motifs in proteins', *J. Mol. Biol.* **229**, 707–721.

He, X., Cai, D., Liu, H. & Ma, W. (2004), Locality preserving indexing for document representation, *in* 'Proceedings of the 27th Annual Internation ACM SIGIR Conference on Research and Development in Information Retrieval', pp. 96–103.

He, X. & Niyogi, P. (2003), Locality preserving projections, *in* 'Advances in Neutral Information Processing Systems 16 $NIPS$2003'.

Holm, L. & Sander, C. (1993), 'Protein structure comparison by alignment of distance matrices', *J. Mol. Biol.* **233**, 123–138.

Holm, L. & Sander, C. (1996), 'Mapping the protein universe', *Science* **273**, 595–603.

Huang, Z. (2005), Indexing protein substructures for efficient similarity queries, Technical report, School of ITEE, University of Queensland, http://www.itee.uq.edu.au/ huang/Report.pdf.

Huang, Z., Zhou, X. & Song, D. (2005), High dimensional indexing for protein structure matching using bowties, *in* 'Proc. of 3rd Asia-Pacific Bioinformatics Conference', pp. 21–30.

Landauer, T., Foltz, P. & Laham, D. (1998), 'Introduction to latent semantic analysis', *Discourse Process* **25**(2&3), 259–284.

Mount, D. (2001), *Bioinformatics: Sequence and Genome Analysis*, Cold Spring Harbor Laboratory Press.

Nussinov, R. & Wolfson, H. (1991), 'Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques.', *Proc. Natl Acad. Sci.* **October-December**, 10495–10499.

Orengo, C., Jones, D. & Thornton, J. (2003), 'Bioinformatics: Genes, proteins and computers'.

Orengo, C. & Taylor, W. (1989), 'Protein structure alignment', *J. Mol. Biol.* **208**, 1–22.

Orengo, C. & Taylor, W. (1996), 'Ssap: Sequential structure alignment program for protein structure comparison', *Methods Enzymol.* **266**, 617–635.

Protein data Bank (n.d.), http://www.rcsb.org/pdb/.

Spriggs, R., Artymiuk, P. & Willett, P. (2003), 'Searching for patterns of amino acids in 3D protein structures', *J Chem Inf Comput Sci.* **43**(2), 412–21.

WHATIF relational database (n.d.), http://www.cmbi.kun.nl/gv/whatif/select/.

Wolfson, H. (1997), 'Geometric hashing: an overview.', *IEEE Comp. Science and Eng.* **October-December**, 10–21.