Structural Alignment based Kernels for Protein Structure Classification

Sourangshu Bhattacharya SOURANGSHU@CSA.IISC.ERNET.IN
Chiranjib Bhattacharyya CHIRU@CSA.IISC.ERNET.IN
Dept. of Computer Science and Automation, Indian Institute of Science, Bangalore - 560 012, India.

Nagasuma Chandra NCHANDRA@SERC.IISC.ERNET.IN
Bioinformatics Center, Indian Institute of Science, Bangalore - 560012, India.

Abstract

Structural alignments are the most widely used tools for comparing proteins with low sequence similarity. The main contribution of this paper is to derive various kernels on proteins from structural alignments, which do not use sequence information. Central to the kernels is a novel alignment algorithm which matches substructures of fixed size using spectral graph matching techniques. We derive positive semi-definite kernels which capture the notion of similarity between substructures. Using these as base more sophisticated kernels on protein structures are proposed. To empirically evaluate the kernels we used a 40% sequence non-redundant structures from 15 different SCOP superfamilies. The kernels when used with SVMs show competitive performance with CE, a state of the art structure comparison program.

1. Introduction

Classification of proteins into different classes of interest, is a problem of fundamental interest in computational biology. Powerful techniques exist for classifying proteins having high sequence similarity. However, these methods are not reliable when sequence similarity falls in the twilight zone (Bourne & Shindyalov, 2003), i.e. is in the range $20 - 30\%$. Developing classification tools for such proteins is an important research issue. Since sequence information is unreliable it is interesting to think about classification tools based on structures alone, and deliberately ignore the amino acid types. This view is taken by most structure alignment algorithms (Eidhammer et al., 2000), which are the most widely accepted methods for protein structure comparison.

In this paper, we explore the idea of designing classifiers based on Support vector machines(SVMs) for proteins having low sequence similarity. More explicitly our goal is to derive positive semi-definite kernels motivated from structural alignments without using sequence information. The hypothesis is that any structure classification technique should capture the notion of similarity defined by structural alignment algorithms. Though there is lot of work on designing kernels on protein sequences (Jaakkola et al., 1999; J.-P. Vert, 2004; Leslie & Kwang, 2004), to the best of our knowledge, there is relatively less work on kernels on protein structures. (Wang & Scott, 2005) is an interesting first step, where both sequence and structure information are used, to define kernels on protein structures.

First, we propose a novel protein structure alignment algorithm by using a spectral projection based similarity measure. We show that naive spectral graph matching (Umeyama, 1988) based techniques fail to produce the correct alignment in many cases due to existence of unmatched residues, called indels, in many similar structures. This drawback is remedied by considering pairs of substructures from each protein.

The main contribution of this paper is to propose kernels on protein structures based on structural alignments. Following the idea of sub-structure matching, we propose novel kernels on protein sub-structures using the spectral projection vector and pairwise distances, and show that a limiting case of the spectral kernel relates to the alignment score on substructures. Combining these kernels on sub-structures, we define various kernels on protein structures. Using

Appearing in *Proceedings of the 24^{th} International Conference on Machine Learning*, Corvallis, OR, 2007. Copyright 2007 by the author(s)/owner(s).

substructure kernels, we also define kernels on protein structures that take any pairwise structural alignment, explicitly into account. Benchmarking experiments are conducted on a sequence non-redundant SCOP dataset. We benchmark our results against CE (Bourne & Shindyalov, 1998), a state of the art structure comparison program on a dataset with low sequence similarity. Extensive experiments show that the results are promising.

The paper is organized as follows. Section 2 describes development of the structure alignment algorithm. Section 3 develops the kernels for over substructures and subsequently over protein structures. Section 4 describes the experimental results.

2. Protein Structure Alignment

2.1. Background

The tertiary structure of a protein is represented by the 3 dimensional coordinates of all atoms present in a given protein chain, with respect to an arbitrary coordinate system. However, following common practice, we consider only C$^{\alpha}$ atoms of each residue. Thus, we represent protein structures as a set of points corresponding to residues in 3 dimensions. So, a protein P is represented as $P = \{p_1, \dots, p_n\}$ where $p_i \in \mathbb{R}^3, 1 \le i \le n$. A *structural alignment* between two proteins P^A and P^B is a 1-1 mapping $\phi : \{i | p_i^A \in \bar{P}^A\} \rightarrow \{j | p_j^B \in \bar{P}^B\}$, where $\bar{P}^A \subseteq P^A$ and $\bar{P}^B \subseteq P^B$. The mapping ϕ defines a set of correspondences between the residues of the two proteins. $|\bar{P}^A| = |\bar{P}^B|$ is called the *length* of the alignment. Given a structural alignment ϕ between 2 structures P^A and P^B, and a transformation $\mathcal{T}$ of structure B onto A, a popular measure of the goodness of superposition is the *root mean square deviation* (RMSD), defined as $RMSD(\phi) = \sqrt{\frac{1}{|\bar{P}^A|} \sum_{p_i^A \in \bar{P}^A} (p_i^A - \mathcal{T}(p_{\phi(i)}^B))^2}$ Given an alignment ϕ, the optimal transformation $\mathcal{T}$, minimizing RMSD, can be computed in closed form using the method described in (Horn, 1987).

However, using RMSD as a measure for evaluating alignments has 2 problems: the optimal transformation $\mathcal{T}$ needs to be computed for every alignment, and RMSD favors alignments of lower lengths which may not capture the total similarity between 2 proteins in presence of noise. An alternate measure, called the *distance RMSD* avoids the calculation of optimal transformation. Distance RMSD for an alignment ϕ is defined as $RMSD_D(\phi) = \sqrt{\frac{1}{|\bar{P}^A|^2} \sum_{p_i^A, p_j^A \in \bar{P}^A} (d_{ij}^A - d_{\phi(i)\phi(j)}^B)^2}$ where, d_{ij}^A is the distance between residues p_i^A and p_j^A. The matrix d is also called the *distance matrix* of a protein structure.

The drawback of preferring smaller lengths was remedied in formulation proposed in DALI (Holm & Sander, 1993). The score function used in DALI is

$$S_{DALI}(\phi) = \sum_{p_i^A, p_j^A \in \bar{P}^A} \left(0.2 - \frac{|d_{ij}^A - d_{\phi(i)\phi(j)}^B|}{\bar{d}_{ij}} \right) \exp\left(- \left(\frac{\bar{d}_{lk}}{20} \right)^2 \right)$$

where $\bar{d}_{ij} = (d_{ij}^A + d_{\phi(i)\phi(j)}^B)/2$. DALI stochastically searches over all alignments for the ϕ which maximizes S_{DALI}. Note that, residues which are spatially far do not contribute much to the DALI score. This is due to the exponentially decreasing function of the average distance used to weight each term. Following this observation, we define the *adjacency matrix* of a protein as $\mathcal{A}_{ij} = e^{\frac{-d_{ij}}{\alpha}}$, $\alpha > 0$. This is an exponentially decreasing function between 0 and 1, making the entries corresponding to far away residues very close to zero. The problem of finding optimal correspondences can be viewed as an weighted graph matching problem (Umeyama, 1988) with weights given by the adjacency matrix values. In the next section, we propose an alternate formulation using spectral graph theoretic techniques.

2.2. Spectral Solution to Protein Structure Alignment

Consider the ideal situation where both protein P^A and P^B have equal number of residues, all of which have a corresponding residue in the other structure. In such a case, the problem of finding optimal alignment between the two proteins is same as finding the optimal permutation of the residues of one of the proteins. Let $\mathcal{A}^A$ and $\mathcal{A}^B$ be the adjacency matrices corresponding to the two proteins. Consider their eigenvalue decompositions $\mathcal{A}^A = \sum_{i=1}^n \lambda_i^A \zeta_i^A (\zeta_i^A)^T$ and $\mathcal{A}^B = \sum_{i=1}^n \lambda_i^B \zeta_i^B (\zeta_i^B)^T$, with terms on the RHS sorted according to decreasing value of λ. It is easy to see that the eigenvectors ζ_i^A and ζ_i^B will be related by the same permutation as the residues of the original proteins. Thus, the correct alignment can be retrieved by calculating the permutation that best matches the entries of ζ_i^A and ζ_i^B.

The adjacency matrix $\mathcal{A}$ of any protein can be approximated by k eigenvectors with an error η, given by: $\|\mathcal{A} - \sum_{i=1}^k \lambda_i \zeta_i \zeta_i^T\|_F^2 = \sum_{j=k+1}^n \lambda_j^2 = \eta^2 \|\mathcal{A}\|_F^2$. For $k = 1$, η is minimized when the eigenvector corresponding to the largest eigenvalue, ζ_1, is considered. The j^{th} component of the leading eigenvector, $\zeta_1(j)$ can be thought of as a projection of the corresponding residue on to the real line. It was also observed that spatially close residues have similar projected values; i.e. if $\mathcal{A}_{jl}$ is high, then $|\zeta_1(j) - \zeta_1(l)|$ is low. Hence, ζ_1

can be thought of as projections that preserve neighborhood (Bhattacharya et al., 2006).

Let ζ_1^A and ζ_1^B be the eigenvectors corresponding to the largest eigenvalues of the adjacency matrices of proteins P^A and P^B. Define *spectral projection vector* f, to be a vector of absolute values of components of ζ. Thus, $f_i^A = |\zeta_1^A(i)|$ and $f_i^B = |\zeta_1^B(i)|$. The absolute values of the eigenvectors are taken because if ζ is an eigenvector, so is $-\zeta$. We define the similarity between residue i of P^A and residue j of P^B as:

$$s(i,j) = T - (f_i^A - f_j^B)^2 \qquad (1)$$

Here, T is a threshold on the minimum similarity between f_i^A and f_J^B for them to be included in an alignment. Putting T to be very high will match the entire structures of two proteins. Using this similarity function, we are interested in finding an alignment $\phi : \{i | p_i^A \in \bar{P}^A\} \rightarrow \{j | p_j^B \in \bar{P}^B\}$ that solves:

$$S(\phi^*) = \max_{\phi} \sum_{i:p_i^A \in \bar{P}^A} s(i, \phi(i)) \qquad (2)$$

This is an instance of the *assignment problem*, or weighted bipartite graph matching problem, which can be solved using linear programming (Bertsimas & Tsitsiklis, 1997).

Unfortunately, in many cases, this technique fails to detect the best alignment (see results). This is due to the fact that many protein structures have a significant number of residues which don't have a corresponding residue in other similar structure. These extra residues, called *indels*, add extra rows and columns to the adjacency matrix, thereby changing the eigenvalues and eigenvectors. This shortcoming of the spectral method can be overcome in case of protein structures by considering sub-structures instead of the whole structures for alignment. The reason is that even though there may be many indels between two similar structures, the portions responsible for function of the protein or for maintaining the protein fold remain highly conserved. Thus, considering sub-structures of equal size (in number of residues) around each residue is likely to give many entirely matched pairs of sub-structures in case of similar proteins, and vice versa.

A sub-structure N_i^A of protein P^A centered at residue i is a set of l residues that are closest to residue i in 3-D space, l being the size of sub-structures being considered. Thus, $N_i^A = \{p_j^A \in P^A | p_k^A \notin N_i^A \Rightarrow \|p_j^A - p_i^A\| \leq \|p_k^A - p_i^A\|$ and $|N_i^A| = l\}$. The robust algorithm for comparing protein structuresP^A and P^B is:

1. Compute the sub-structures centered at each residue for both proteins P^A and P^B.

2. For each pair of sub-structures, one from each protein, compute the alignment between the sub-structures by solving the problem described in equations 1 and 2.

3. For each sub-structure alignment computed above, compute the optimal transformation of the sub-structure from P^A onto the one from P^B. Transform the whole of P^A onto P^B using the computed transformation, and compute the similarity score between residues of P^A and P^B.

4. Compute the optimal alignment between P^A and P^B by solving the assignment problem described in equation 2 using similarity score computed above.

5. Report the best alignment of all the alignments computed in the above step as the optimal one.

The sub-structure based algorithm relies on the assumption that the optimal structural alignment between two protein structures contains at least one pair of optimally and fully aligned sub-structures, one from each of the proteins. For each sub-structure alignment computed in step 2 of the above algorithm, the optimal transformation superposing the corresponding residues is calculated using the method described in (Horn, 1987). The "best" alignment in mentioned in step 5 is decided on the basis of both RMSD and the length of alignment. A detailed description and benchmarking of the method will be presented elsewhere. Taking ideas from the algorithm designed in this section, we propose kernels for proteins structures in the next section.

3. Kernels for Protein Structure Classification

3.1. Kernels for Sub-structures

The algorithm designed in the previous section motivates the notion of first proposing kernels on sub-structures and then suitably combining them to derive kernels on entire protein structures. To this end, we propose a kernel over sub-structures in this section. (Wang & Scott, 2005) also combine substructure kernels to build kernels on protein structures. However, their kernels use amino acid similarity inside substructures. Also, since a kernel function captures the similarity between two objects, the algorithm gives us the idea of using the spectral projection vectors for each sub-structure to define the substructure kernels.

Let N_1 and N_2 be the sub-structures, each having l residues. Let f^1 and f^2 be the spectral projection vectors for each sub-structure. We define the kernel $\mathcal{K}_{res}$ between i^{th} residue of N_1 and j^{th} residue of N_2 as the decreasing function of the difference in spectral projection: $\mathcal{K}_{res}(i,j) = e^{\frac{-(f_i^1 - f_j^2)^2}{\beta}}$. Technically, since a kernel should be defined over a set of objects, we extend this definition to the union of the sets of residues from the two sub-structures, though only the cross terms appear in subsequent calculations. It is easy to see that over the union set, this kernel is symmetric and positive semidefinite. Next we combine the kernels over residues of the two sub-structures to define kernels over sub-structures using the convolution kernel technique (Haussler, 1999).

Convolution kernels were proposed in (Haussler, 1999) as a general tool for building kernels on complex objects formed by combining simple objects, on which kernels are already known. Let $x \in X$ be a composite object formed using parts from $X_1, \ldots, X_m$. Let R be a relation over $X_1 \times \cdots \times X_m \times X$ such that $R(x_1, \ldots, x_m, x)$ is true if x is composed of $x_1, \ldots, x_m$. Let $R^{-1}(x) = (x_1, \ldots, x_m) \in X_1 \times \cdots \times X_m | R(x_1, \ldots, x_m, x) = \text{true}$ and $K^1, \ldots, K^m$ be kernels on $X_1, \ldots, X_m$, respectively. The convolution kernel K over X is defined as:

$$K(x,y) = \sum_{(x_1,\ldots,x_m)\in R^{-1}(x),(y_1,\ldots,y_m)\in R^{-1}(y)} \prod_{i=1}^{m} K^i(x_i, y_i) \tag{3}$$

As shown in (Haussler, 1999), if $K^1, \ldots, K^m$ are symmetric and positive semidefinite, so is K.

Sub-structure Spectral Kernel. In our case, X is the set of all sub-structures and $X_1, \ldots, X_m$ are all sets of all the residues p_i's from all the sub-structures. Here, note that even if the same residue appears in different sub-structures, the appearances will be considered to be different. Since, each sub-structure has a size of l, in our case $m = l$. The relation R is defined as $R(p_1, \ldots, p_l, N)$ is true iff $N = \{p_1, \ldots, p_l\}$. Since, all the X_i's have all the residues from N, the cases for which R can hold true are the permutations of residues of N. Since, this can happen for both N_1 and N_2, each combination of correspondences occurs $l!$ times. Thus, from the above equation, the kernel becomes: $\mathcal{K}(N_1, N_2) = l! \sum_{\pi\in\Pi(l)} e^{\frac{1}{\beta} - \sum_{k=1}^{l}(f_k^1 - f_{\pi(k)}^2)^2} = l! \sum_{\pi\in\Pi(l)} e^{\frac{-\|f^1 - \pi(f^2)\|^2}{\beta}}$ where, f^i is the spectral projection vector of N_i and $\Pi(l)$ is the set of all possible permutations of l numbers. Since $l!$ is a constant, we define the *sub-structure spectral* kernel as:

$$\mathcal{K}_{SS}(N_i, N_j) = \sum_{\pi\in\Pi(l)} e^{\frac{-\|f^i - \pi(f^j)\|^2}{\beta}} \tag{4}$$

Pairwise distance substructure kernel. Using the convolution kernel technique, we define another kernel on substructures based on pairwise distances between residues. In this case, X is the set of all sub-structures and $X_1, \ldots, X_m$ are all sets of all pairwise distances $d_{ij},\ i < j$ between the residues from all substructures. Thus, in this case $m = l(l-1)/2$. The relation R is defined as $R(d_{12}, \ldots, d_{(l-1),l}, N)$ is true iff $N = \{p_1, \ldots, p_l\}$ and $d_{ij} = \|p_i - p_j\|$. Thus, R holds true for $\mathbf{d} = (d_1, \ldots, d_m)$ and N, if $\mathbf{d}$ is the pairwise distance vector of N for all permutations of indices of residues. So, with the notation that $(\pi(\mathbf{d}))_{i,j} = \|p_{\pi(i)} - p_{\pi(j)}\|$, the *pairwise distances substructure* kernel is defined as:

$$\mathcal{K}_{PDS}(N_i, N_j) = \sum_{\pi\in\Pi(l)} e^{\frac{-\|\mathbf{d}^i - \pi(\mathbf{d}^j)\|^2}{\sigma^2}} \tag{5}$$

Using the result in (Haussler, 1999), $\mathcal{K}_{SS}$ and $\mathcal{K}_{PDS}$ are positive semidefinite. Next, we prove a relation between the score obtained by solving the assignment problem in equation 2 and $\mathcal{K}_{SS}$.

Theorem 3.1 *Let N_i and N_j be two sub-structures with spectral projection vectors f^i and f^j. Let $S(N_i, N_j)$ be the score of alignment of N_i and N_j, obtained by solving the assignment problem specified in equation 2, for large enough value of T such that all residues are matched.*

$$e^{lT} \lim_{\beta\to 0} \mathcal{K}_{SS}(N_i, N_j))^{\beta} = e^{S(N_i,N_j)}$$

Proof: Let π^* be the permutation that gives the optimal score $S(N_i, N_j)$. By definition, $e^{S(N_i,N_j)} = \max_{\pi\in\Pi(l)} e^{\sum_{k=1}^{l} T - (f_k^i - f_{\pi(k)}^j)^2} = e^{lT} e^{-\|f^i - \pi^*(f^j)\|^2}$.

$$\begin{aligned}
&\lim_{\beta\to 0}(\mathcal{K}_{SS}(N_i, N_j))^{\beta} \\
&= \lim_{\beta\to 0}(\textstyle\sum_{\pi\in\Pi(l)} e^{\frac{-\|f^i - \pi(f^j)\|^2}{\beta}})^{\beta} \\
&= e^{-\|f^i - \pi^*(f^j)\|^2} \lim_{\beta\to 0}(1 + \textstyle\sum_{\pi\in\Pi(l)\setminus\{\pi^*\}} T_1)^{\beta} \\
&= e^{-\|f^i - \pi^*(f^j)\|^2}
\end{aligned}$$

where, $T_1 = e^{\frac{-1}{\beta}(\|f^i - \pi(f^j)\|^2 - \|f^i - \pi^*(f^j)\|^2)}$.

A similar result holds for $\mathcal{K}_{PDS}$. However, in that case optimizing the similarity measure is an NP-complete problem (Pardalos et al., 1994). A similar result was proved for the local alignment kernel defined on strings in (J.-P. Vert, 2004). We define the limiting case of $\mathcal{K}_{SS}$ as a new kernel $\mathcal{K}_{LSS}(N_1, N_2) = \lim_{\beta\to 0}(\mathcal{K}_{SS}(N_1, N_2))^{\beta}$.

Calculation of $\mathcal{K}_{SS}$ and $\mathcal{K}_{PDS}$ takes time exponential in the substructure size. However, it is not a major hindrance as the substructure size is fixed to a small number (6 in our experiments). $\mathcal{K}_{LSS}$ can be computed in time polynomial in substructure size. Unfortunately, $\mathcal{K}_{LSS}$ is not necessarily positive semidefinite. However, in the next section, we develop positive semidefinite kernels on protein structures using $\mathcal{K}_{SS}$, $\mathcal{K}_{PDS}$ and $\mathcal{K}_{LSS}$.

3.2. Kernels on Protein Structures

Viewing kernel values as a measure of similarity between the proteins, it is natural to consider the sum of sub-structure kernels over all pairs of sub-structures between the two proteins as a kernel between the proteins. Thus, given proteins $P^i = \{p^i_1, \ldots, p^i_{n_i}\}$ and $P^j = \{p^j_1, \ldots, p^j_{n_j}\}$, we define the following two kernel based on the two sub-structure kernels defined earlier:

$$\mathcal{K}_1(P^i, P^j) = \sum_{a=1}^{n_i}\sum_{b=1}^{n_j} \mathcal{K}_{SS}(N^i_a, N^j_b) \qquad (6)$$

$$\mathcal{K}_2(P^i, P^j) = \sum_{a=1}^{n_i}\sum_{b=1}^{n_j} \mathcal{K}_{PDS}(N^i_a, N^j_b) \qquad (7)$$

Here, N^i_a is the sub-structure in protein P^i centered at residue p^i_a. Both $\mathcal{K}_1$ and $\mathcal{K}_2$ are positive semidefinite because they are sum of positive semidefinite kernels.

The kernels defined above add up the substructure kernel values for all possible pairs of sub-structures. However, this is bad estimate of the similarity between the proteins as most of the sub-structure pairs even for similar proteins might not be similar. One possible way of increasing the accuracy is to consider two substructures from each protein and weight the substructure kernels with a decreasing function of the distance between them. To maintain positive semi-definiteness of the resulting function, it is necessary that the weighting function be positive semidefinite. We use the gaussian kernel, $\mathcal{K}_{norm}((N^i_a, N^i_b), (N^j_c, N^j_d)) = e^{\frac{(\|p^i_a - p^i_b\| - \|p^j_c - p^j_d\|)}{\sigma^2}}$, as the weighting function. Note that in this case the kernel is over all pairs of substructures in each structure. Thus, we define two more kernels as:

$$\mathcal{K}_3(P^i, P^j) = \sum_{a,b=1}^{n_i}\sum_{c,d=1}^{n_j} \mathcal{K}_{SS}(N^i_a, N^i_b) \times \mathcal{K}_{SS}(N^j_c, N^j_d) \times \mathcal{K}_{norm}((N^i_a, N^i_b), (N^j_c, N^j_d)) \qquad (8)$$

$$\mathcal{K}_4(P^i, P^j) = \sum_{a,b=1}^{n_i}\sum_{c,d=1}^{n_j} \mathcal{K}_{PDS}(N^i_a, N^i_b) \times \mathcal{K}_{SS}(N^j_c, N^j_d) \times \mathcal{K}_{norm}((N^i_a, N^i_b), (N^j_c, N^j_d)) \qquad (9)$$

It is easy to see that $\mathcal{K}_3(P^i, P^j)$ and $\mathcal{K}_4(P^i, P^j)$ are both positive semidefinite. Even though these kernels are more accurate than the previous two, they computationally expensive. Computation time of these kernels is $O(n^4)$ which is also more than that of many structural alignment algorithms (Bourne & Shindyalov, 1998; Bhattacharya et al., 2006).

Another interesting way of increasing accuracy of kernels is by taking correspondences produced by a structural alignment algorithm into account while computing the kernel values. Thus, given an alignment ϕ_{ij} between proteins P^i and P^j we define the *alignment kernels* as :

$$\mathcal{K}^{Al}_1(P^i, P^j; \phi_{ij}) = \sum_{a|p^i_a \in \bar{P}^i} \mathcal{K}_{SS}(N^i_a, N^j_{\phi_{ij}(a)}) \qquad (10)$$

$\mathcal{K}^{Al}_2$ and $\mathcal{K}^{Al}_3$ are defined analogously by replacing $\mathcal{K}_{SS}$ by $\mathcal{K}_{LSS}$ and $\mathcal{K}_{PDS}$ in the above equation, respectively. Unfortunately, these kernels are not necessarily positive semidefinite. For such cases, a standard trick is to compute the eigenvector decomposition of the kernel matrix, force all the negative eigenvalues to be zero and recompute the kernel matrix from the modified eigenvector decomposition. In experiments, this trick is used to make these kernels positive semidefinite.

However, this requires computation of the eigenvalues of the kernel matrix. We propose another set of kernel that while keeping the off-diagonal terms intact and only modifying the diagonal terms, is always positive semidefinite. Thus, given a dataset of proteins $\{P^1, \ldots, P^M\}$, and all possible pairwise alignments ϕ_{ij} between proteins P^i and P^j, we define the kernel $\mathcal{K}_3$ as:

$$\mathcal{K}^{Al}_4(P^i, P^j) = \begin{cases} \displaystyle\sum_{a|p^i_a \in \bar{P}^i} \mathcal{K}_{SS}(N^i_a, N^j_{\phi_{ij}(a)}) & \text{if } i \neq j \\ \displaystyle\sum_{b=1}^{M} \sum_{a|p^i_a \in \mathrm{dom}(\phi_{ib})} \mathcal{K}_{SS}(N^i_a, N^i_{\phi_{ib}(a)}) & \text{if } i = j \end{cases} \qquad (11)$$

where, $\mathrm{dom}(\phi_{ib})$ is the domain of function ϕ_{ib}, which is the set of all residues participating in the alignment ϕ_{ib} from structure P^i. $\mathcal{K}^{Al}_5$ and $\mathcal{K}^{Al}_6$ are defined analogously by replacing $\mathcal{K}_{SS}$ with $\mathcal{K}_{LSS}$ and $\mathcal{K}_{PDS}$, respectively.

Theorem 3.2 *$\mathcal{K}^{Al}_4$ is positive semidefinite if $\mathcal{K}_{SS}$ is positive valued.*

Proof: Let L_{max} be the maximum length of alignments between all pairs of proteins in the dataset. Consider the $M \times (\frac{M(M+1)}{2} L_{max})$ matrix H having one row for each proteins in the dataset. Each row has $\frac{M(M+1)}{2}$ block of length L_{max} corresponding to each pairwise alignment (including alignment of every structure with itself). For rows corresponding to proteins i and j, the k^{th} element of the block for alignment ϕ_{ij} is equal to $\sqrt{\mathcal{K}_{SS}(N^i_k, N^j_{\phi_{ij}(k)})}$. The index k runs over all the correspondences in the alignment. For alignments that have length smaller than L_{max}, put the remaining entries to be zero. It can be seen that $\mathcal{K}_4^{Al} = HH^T$. As, each entry in $\mathcal{K}_4^{Al}$ is a dot product between two vectors, $\mathcal{K}_4^{Al}$ is positive semidefinite.

Note that $\mathcal{K}_{SS}$ needs to be positive valued and not positive semidefinite. So, $\mathcal{K}_5^{Al}$ and $\mathcal{K}_6^{Al}$ are also positive semidefinite, even though $\mathcal{K}_{LSS}$ is not. Computational complexity of $\mathcal{K}_1$ and $\mathcal{K}_2$ is $O(n_1 n_2)$ and that of $\mathcal{K}_3$ and $\mathcal{K}_4$ is $O(n_1^2 n_2^2)$. The alignment kernels have complexity of $O(N_{al})$, where N_{al} is the number of aligned residues. In the next section, we report results of experiments conducted for validating the structure alignment algorithm and the kernels developed in this section for protein structure classification.

4. Experimental Results

The algorithms and kernels developed above were implemented and tested on real protein structures from various structural classes obtained from PDB (Berman et al., 2000). The alignment algorithms and kernels were implemented in C using GCC/GNU Linux. Eigenvalue computations were done using Lapack [1]. SVM classifications were done using Libsvm [2].

Alignment algorithms were validated against their popularly used counterpart e.g. DALI (Holm & Sander, 1993) and CE (Bourne & Shindyalov, 1998). Due to lack of space, only indicative results are reported here. Kernels were evaluated on the the task of classifying protein structures. SCOP (Murzin et al., 1995) was taken as the standard for protein structure classification. Nearest neighbor classification using zscore given by CE (Bourne & Shindyalov, 1998) was used as the standard method.

4.1. Structural Alignment Results from Spectral Projection based Algorithms

In order to test the protein structure alignment algorithms described in section 2.2, we used the algorithms to compare a number of similar protein structures retrieved from the PDB. Two other popularly used protein structure algorithms, e.g. DALI (Holm & Sander, 1993) and CE (Bourne & Shindyalov, 1998) were also used to compare the same proteins. Table 1 shows the RMSDs and lengths of alignment given by the programs for a set of proteins from a variety of classes. The results show that on similar proteins both the spectral methods perform at par with the existing techniques.

On the pair `2PEL - 5CNA` the spectral algorithms give much better alignments than the existing methods. This is due to the fact that this pair of proteins is related by circular permutations. Also, in order to test the performance of the algorithms on proteins showing indels, we aligned the individual domains of the multidomain protein `2HCK` with the individual domains (table 1). It is clear that the algorithm calculating similarity from spectral projections of the entire protein structure doesn't report the correct alignments even when the sub-structures are highly similar (RMSD 0). Thus, it is clear that the sub-structure based method is more robust to indels than the naive spectral method.

4.2. Structure Classification using Kernels

In order to test the effectiveness of kernels proposed in this article, we consider a difficult task of classifying proteins with low sequence similarity, the SCOP (Murzin et al., 1995). SCOP provides a 40% sequence non-redundant dataset (the lowest cutoff provided by SCOP), having about 4000 proteins. The entire dataset was taken, and superfamilies having more than 10 proteins were found (total 21 of them). Of these 15 superfamilies were taken and 10 proteins from each superfamily were chosen randomly.

Experiments were performed on this dataset, using a protocol similar to that in (Wang & Scott, 2005). 15 binary classification problems were posed for identifying each class, where the negative data contained 10 proteins (to keep the dataset balanced) randomly chosen from all other classes. Leave one out cross validation using SVM was performed on all 15 of such classification problems, and positive and negative error rates are reported in table 2. The experiments were repeated 10 times with different negative datasets, and the average accuracy was reported. Alignments required for the alignment kernels were computed using the method described in (Bhattacharya et al., 2006). Also, the alignment kernels ($\mathcal{K}_4^{Al} - \mathcal{K}_6^{Al}$), showed the problem of diagonal dominance, which was remedied using the method suggested in (Schölkopf et al., 2002). Results reported for CE (Bourne & Shindyalov, 1998) are same experiments, except that nearest neighbor classifier was used taking zscore values as the similarity measure.

Kernels $\mathcal{K}_1$ and $\mathcal{K}_2$ perform reasonably well in terms of positive classification accuracy, though on an average, negative classification accuracy is lower. $\mathcal{K}_3$ and $\mathcal{K}_4$ also show similar performance except on the last class (SCOP superfamily a.39.1). Performance of $\mathcal{K}_3$ and $\mathcal{K}_4$ did not achieve the expected improvement due to inadequate tuning of parameters. In general, the

[1] Available at: http://www.netlib.org/clapack/

[2] Available at: http://www.csie.ntu.edu.tw/c̃jlin/libsvm

alignment kernels $\mathcal{K}_1^{Al} - \mathcal{K}_6^{Al}$, show better performance than other kernels. Also, out of the alignment kernels, the positive semidefinite kernels ($\mathcal{K}_4^{Al} - \mathcal{K}_6^{Al}$) show better performance than the non-positive semidefinite ones ($\mathcal{K}_1^{Al} - \mathcal{K}_3^{Al}$), as expected. Finally, the positive semidefinite alignment kernels ($\mathcal{K}_4^{Al} - \mathcal{K}_6^{Al}$) show better overall classification accuracy than CE, a state of the art structure comparison tool. Interestingly, CE shows much lower negative classification accuracy than many of the kernels. This can be attributed to better generalization capabilities of SVMs over nearest neighbor classifiers. Thus, it is clear from experiments that kernels based alignment outperform, state of the art structure classification tools in terms of accuracy.

Kernels $\mathcal{K}_1$, $\mathcal{K}_2$ take less than $5s$, and $\mathcal{K}_3$, $\mathcal{K}_4$ take less than $30s$ for proteins having around 200 residues on an Athlon 2.2GHz based desktop computer. The alignment kernels take less than $1s$ for 200 residue proteins.

5. Conclusion

In this study, the main objective was to derive kernels on protein structures, without using sequence information. Several kernels were derived from structural alignments, and were benchmarked against CE, on the difficult task classifying proteins having low sequence similarity, with extremely encouraging results. Though the kernels were designed on protein structures, it is interesting to explore it's applications to other 3D pointsets.

Acknowledgement: The authors are indebted to MHRD, Govt. of India, for supporting this research throught grant number F26-11/2004.

References

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., & Bourne, P. E. (2000). The protein data bank. *Nucleic Acids Research*, *28*, 235–242.

Bertsimas, D., & Tsitsiklis, J. (1997). *Introduction to linear optimization*. Athena Scientific.

Bhattacharya, S., Bhattacharyya, C., & Chandra, N. (2006). Projections for fast protein structure retrieval. *BMC Bioinformatics*, *7 suppl.*, 5:S5.

Bourne, P. E., & Shindyalov, I. N. (1998). Protein structure alignment by incremental combinatorial extension of optimal path. *Protein Engineering*, *11*, 739–747.

Bourne, P. E., & Shindyalov, I. N. (2003). Protein structure comparison and alignment. In P. E. Bourne and H. Weissig (Eds.), *Structural bioinformatics*, 321–337. Wiley-Liss.

Eidhammer, I., Jonassen, I., & Taylor, W. R. (2000). Structure comparison and structure patterns. *Journal of Computational Biology*, *7*, 685–716.

Haussler, D. (1999). *Convolution kernels on discrete structures* (Technical Report). University of California, Santa Cruz.

Holm, L., & Sander, C. (1993). Protein structure comparison by alignment of distance matrices. *Journal of Molecular Biology*, *233*, 123–138.

Horn, B. K. P. (1987). Closed form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, *4*, 629–642.

J.-P. Vert, H. Saigo, T. A. (2004). *Kernel methods in computational biology*, chapter Local alignment kernels for biological sequences, 131–154. MIT Press.

Jaakkola, T., Diekhaus, M., & Haussler, D. (1999). Using the fisher kernel method to detect remote protein homologies. *7th Intell. Sys. Mol. Biol.*, 149–158.

Leslie, C., & Kwang, R. (2004). Fast string kernels using inexact matching for protein sequences. *Journal of Machine Learning Research*, *5*, 1435 − 1455.

Murzin, A. G., Brenner, S. E., Hubbard, T., & Chothia, C. (1995). Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, *247*, 536–540.

Pardalos, P., Rendl, F., & Wolkowicz, H. (1994). The quadratic assignment problem: a survey and recent developments. In P. Pardalos and H. Wolkowicz (Eds.), *Quadratic assignment and related problems (new brunswick, NJ, 1993)*, 1–42. Providence, RI: Amer. Math. Soc.

Schölkopf, B., Weston, J., Eskin, E., Leslie, C. S., & Noble, W. S. (2002). A kernel approach for learning from almost orthogonal patterns. *ECML* (pp. 511–528).

Umeyama, S. (1988). An eigendecomposition approach to weighted graph matching problems. *IEEE transactions on pattern analysis and machine intelligence*, *10*, 695–703.

Wang, C., & Scott, S. D. (2005). New kernels for protein structural notif discovery and function classification. *International Conference on Machine Learning*.

Table 1. Comparison of results for pairwise protein structure comparison from different programs

Protein1 (length)	Protein2 (length)	C.E. RMSD (N)	DALI RMSD (N)	Spectral (full prot.) RMSD (N)	Spectral (robust) RMSD (N)
1DWT:A (152)	2MM1 (153)	0.7 (152)	0.8 (152)	0.749 (152)	0.74 (152)
5CNA:A (237)	2PEL:A (232)	1.2 (115)	1.3 (117)	1.911 (223)	1.33 (221)
2ACT (218)	1PPN (212)	1.0 (211)	0.9 (210)	0.877 (210)	0.82 (211)
1HTI:A (248)	1TIM:A (247)	0.9 (246)	1.0 (247)	1.023 (247)	0.86 (245)
2HCK:A (437)	d2hcka1 (63)	-	0.0 (63)	2.81 (34)	0.0 (63)
2HCK:A (437)	d2hcka2 (103)	-	0.0 (103)	3.12 (58)	0.0 (103)
2HCK:A (437)	d2hcka3 (271)	-	0.0 (271)	0.00 (271)	0.0 (271)

Table 2. Positive and negative classification accuracies on 15 SCOP classes for various kernels and CE

SCOP	$\mathcal{K}_1$		$\mathcal{K}_2$		$\mathcal{K}_3$		$\mathcal{K}_4$		$\mathcal{K}_1^{Al}$		$\mathcal{K}_2^{Al}$	
Classfn	TP %	TN %	TP %	TN %	TP %	TN %	TP %	TN %	TP %	TN %	TP %	TN %
a.4.5	77.00	66.00	62.00	75.00	55.00	85.00	82.00	87.00	94.00	89.00	91.00	97.00
c.66.1	63.00	45.00	72.00	38.00	81.00	49.00	55.00	54.00	83.00	100.00	97.00	100.00
b.18.1	92.00	78.00	89.00	81.00	74.00	50.00	98.00	69.00	87.00	92.00	90.00	82.00
c.52.1	67.00	52.00	72.00	50.00	58.00	47.00	47.00	49.00	35.00	95.00	43.00	80.00
c.37.1	51.00	36.00	58.00	36.00	35.00	47.00	52.00	44.00	33.00	86.00	42.00	78.00
b.29.1	100.00	87.00	87.00	71.00	82.00	51.00	80.00	81.00	70.00	98.00	79.00	95.00
c.108.1	66.00	45.00	74.00	50.00	70.00	40.00	81.00	59.00	65.00	100.00	79.00	99.00
c.47.1	63.00	44.00	35.00	38.00	52.00	63.00	54.00	63.00	81.00	90.00	81.00	82.00
b.1.18	72.00	65.00	70.00	75.00	49.00	75.00	63.00	81.00	93.00	49.00	91.00	85.00
g.39.1	59.00	45.00	54.00	86.00	60.00	97.00	76.00	95.00	100.00	15.00	96.00	57.00
b.40.4	65.00	72.00	77.00	76.00	53.00	87.00	68.00	83.00	87.00	58.00	90.00	83.00
c.55.3	62.00	30.00	65.00	39.00	68.00	42.00	60.00	56.00	60.00	99.00	68.00	96.00
c.55.1	57.00	38.00	58.00	47.00	33.00	50.00	54.00	35.00	58.00	97.00	65.00	85.00
c.2.1	68.00	45.00	65.00	64.00	42.00	41.00	60.00	47.00	84.00	93.00	87.00	91.00
a.39.1	83.00	75.00	93.00	94.00	30.00	0.00	30.00	11.00	85.00	91.00	88.00	87.00

Contd...

SCOP	$\mathcal{K}_3^{Al}$		$\mathcal{K}_4^{Al}$		$\mathcal{K}_5^{Al}$		$\mathcal{K}_6^{Al}$		CE	
Classfn	TP %	TN %	TP %	TN %	TP %	TN %	TP %	TN %	TP %	TN %
a.4.5	93.00	93.00	83.00	91.00	82.00	90.00	90.00	92.00	93.00	60.00
c.66.1	81.00	100.00	99.00	86.00	90.00	85.00	100.00	90.00	99.00	45.00
b.18.1	94.00	78.00	80.00	88.00	82.00	87.00	98.00	84.00	100.00	77.00
c.52.1	31.00	82.00	60.00	62.00	65.00	64.00	76.00	65.00	86.00	54.00
c.37.1	27.00	89.00	70.00	50.00	66.00	53.00	76.00	67.00	93.00	41.00
b.29.1	66.00	96.00	84.00	66.00	86.00	69.00	97.00	82.00	97.00	74.00
c.108.1	69.00	100.00	81.00	69.00	84.00	66.00	92.00	80.00	100.00	69.00
c.47.1	68.00	89.00	64.00	71.00	61.00	68.00	72.00	71.00	98.00	58.00
b.1.18	96.00	45.00	81.00	80.00	84.00	81.00	85.00	82.00	99.00	78.00
g.39.1	100.00	37.00	90.00	91.00	84.00	89.00	94.00	86.00	85.00	85.00
b.40.4	92.00	61.00	75.00	78.00	67.00	78.00	80.00	78.00	98.00	70.00
c.55.3	59.00	98.00	89.00	66.00	80.00	67.00	98.00	84.00	100.00	56.00
c.55.1	57.00	91.00	89.00	69.00	91.00	69.00	85.00	72.00	99.00	52.00
c.2.1	92.00	98.00	100.00	89.00	95.00	87.00	95.00	78.00	100.00	57.00
a.39.1	83.00	83.00	88.00	83.00	93.00	88.00	90.00	92.00	100.00	74.00