fRMSDPred: Predicting local rmsd between structural fragments using sequence information

Huzefa Rangwala and George Karypis

Computer Science & Engineering, University of Minnesota, Minneapolis, MN 55455
rangwala@cs.umn.edu, karypis@cs.umn.edu

UMN-CSE Technical Report 07–011

Abstract

The effectiveness of comparative modeling approaches for protein structure prediction can be substantially improved by incorporating predicted structural information in the initial sequence-structure alignment. Motivated by the approaches used to align protein structures, this paper focuses on developing machine learning approaches for estimating the RMSD value of a pair of protein fragments. These estimated fragment-level RMSD values can be used to construct the alignment, assess the quality of an alignment, and identify high-quality alignment segments.

We present algorithms to solve this fragment-level RMSD prediction problem using a supervised learning framework based on support vector regression and classification that incorporates protein profiles, predicted secondary structure, effective information encoding schemes, and novel second-order pairwise exponential kernel functions. Our comprehensive empirical study shows superior results compared to the profile-to-profile scoring schemes.

Keywords: *structure prediction, comparative modeling, machine learning, classification, regression*

1 Introduction

Over the years, several computational methodologies have been developed for determining the 3D structure of a protein (target) from its linear chain of amino acid residues [27, 12, 32, 23, 31, 28]. Among them, approaches based on comparative modeling [27, 28] are the most widely used and have been shown to produce some of the best predictions when the target has some degree of homology with proteins of known 3D structure (templates) [3, 42].

The key idea behind comparative modeling approaches is to align the sequence of the target to the sequence of one or more template proteins and then construct the target's structure from the structure of the template(s) using the alignment(s) as a reference. Thus, the construction of high-quality target-template alignments plays a critical role in the overall effectiveness of the method, as it is used to both select the suitable template(s) and to build good reference alignments. The overall performance of comparative modeling approaches will be significantly improved, if the target-template alignment constructed by considering sequence and sequence-derived information is as close as possible to the structure-based alignment between these two proteins. The development of increasingly more sensitive target-template alignment algorithms [1, 22, 25], that incorporate profiles [7, 2], profile-to-profile scoring functions [5, 17, 39, 8], and predicted secondary structure information [13, 24] have contributed to the continuous success of comparative modeling [37, 38].

The dynamic-programming-based algorithms [19, 33] used in target-template alignment are also used by many methods to align a pair of protein structures. However, the key difference between these two problem settings is that, while the target-template alignment methods score a pair of aligned residues using sequence-derived information, the structure alignment methods use information derived from the structure of the protein. For example, structure alignment methods like CE [30] and MUSTANG [15] score a pair of residues by considering how well fixed-length fragments (i.e., short contiguous backbone segments) centered around each residue align with each other. This score is usually computed as the root mean squared deviation (RMSD) of the optimal superimposition of the two fragments.

In this paper, motivated by the alignment requirements of comparative modeling approaches and the operational characteristics of protein structure alignment algorithms, we focus on the problem of estimating the RMSD value of a pair of protein fragments by considering only sequence-derived information. Besides its direct application to target-template alignment, accurate estimation of these fragment-level RMSD values can also be used to solve a number of other problems related to protein structure prediction such as identifying the best template by assessing the quality of target-template alignments and identifying high-quality segments of an alignment.

We present algorithms to solve the fragment-level RMSD prediction problem using a supervised learning framework based on support vector regression and classification that incorporates sequence-derived information in the form of position-specific profiles and predicted secondary structure [14]. This information is effectively encoded in fixed-length feature vectors. We develop and test novel second-order pairwise exponential kernel functions designed to capture the conserved signals of a pair of local windows centered at each of the residues and use a fusion-kernel-based approach to incorporate the profile- and secondary structure-based information.

An extensive experimental evaluation of the algorithms and their parameter space is performed using a dataset of residue-pairs derived from optimal sequence-based local alignments of known protein structures. Our experimental results show that there is a high correlation (0.681 – 0.768) between the estimated and actual fragment-level RMSD scores. Moreover, the performance of our algorithms is considerably better than that obtained by state-of-the-art profile-to-

profile scoring schemes when used to solve the fragment-level RMSD prediction problems.

The rest of the paper is organized as follows. Section 2, provides key definitions and notations used throughout the paper. Section 3 formally defines the fragment-level RMSD prediction and classification problems and describes their applications. Section 4 describes the prediction methods that we developed. Section 5 describes the datasets and the various computational tools used in this paper. Section 6 presents a comprehensive experimental evaluation of the methods developed. Section 7 summarizes some of the related research in this area. Finally, Section 8 summarizes the work and provides some concluding remarks.

2 Definitions and Notations

Throughout the paper we will use X and Y to denote proteins, x_i to denote the ith residue of X, and $\pi(x_i, y_j)$ to denote the *residue-pair* formed by residues x_i and y_j.

Given a protein X of length n and a user-specified parameter w, we define $w\texttt{mer}(x_i)$ to be the $(2w+1)$-length contiguous subsequence of X centered at position i $(w < i \leq n - w)$. Similarly, given a user-specified parameter v, we define $v\texttt{frag}(x_i)$ to be the $(2v+1)$-length contiguous substructure of X centered at position i $(v < i \leq n - v)$. These substructures are commonly referred to as fragments [30, 15]. Without loss of generality, we represent the structure of a protein using the C_α atoms of its backbone. The $w\texttt{mers}$ and $v\texttt{frags}$ are fixed-length windows that are used to capture information about the sequence and structure around a particular sequence position, respectively.

Given a residue-pair $\pi(x_i, y_j)$, we define $f\texttt{RMSD}(x_i, y_j)$ to be the *structural similarity score* between $v\texttt{frag}(x_i)$ and $v\texttt{frag}(y_j)$. This score is computed as the root mean square deviation between the pair of substructures after optimal superimposition. A residue-pair $\pi(x_i, y_j)$ will be called *reliable* if its $f\texttt{RMSD}$ is bellow a certain value (i.e., there is a good structural superimposition of the corresponding substructures).

Finally, we will use the notation $\langle a, b \rangle$ to denote the dot-product operation between vectors a and b.

3 Problem Statement

The work in this paper is focused on solving the following two problems related to predicting the local structural similarity of residue-pairs.

Definition 1 (fRMSD Estimation Problem) *Given a residue-pair $\pi(x_i, y_j)$, estimate the $f\texttt{RMSD}(x_i, y_j)$ score by considering information derived from the amino acid sequence of X and Y.*

Definition 2 (Reliability Prediction Problem) *Given a residue-pair $\pi(x_i, y_j)$, determine whether it is reliable or not by considering only information derived from the amino acid sequence of X and Y.*

It is easy to see that the reliability prediction problem is a special case to the $f\texttt{RMSD}$ estimation problem. As such, it may be easier to develop effective solution methods for it and this is why we consider it as a different problem in this paper.

The effective solution to these two problems has four major applications to protein structure prediction. First, given an existing alignment between a (target) protein and a template, a prediction of the $f\texttt{RMSD}$ scores of the aligned residue-pairs (or their reliability) can be used to assess the quality of the alignment and potentially select among different alignments and/or different templates. Second, $f\texttt{RMSD}$ scores (or reliability assessments) can be used to analyze different protein-template alignments in order to identify high-quality moderate-length fragments. These fragments can then be used by fragment-assembly-based protein structure prediction methods like TASSER [41] and ROSETTA [26] to construct the structure of a protein. Third, since residue-pairs with low $f\texttt{RMSD}$ scores are good candidates for alignment, the predicted $f\texttt{RMSD}$ scores can be used to construct a position-to-position scoring matrix between all pairs of residues in a protein and a template. This scoring matrix can then be used by an alignment algorithm to compute a high-quality alignment for structure prediction via comparative modeling. Essentially, this alignment scheme uses predicted $f\texttt{RMSD}$ scores in an attempt to mimic the approach used by various structural alignment methods [15, 30]. Fourth, the $f\texttt{RMSD}$ scores (or reliability assessments) can be used as input to other prediction tasks such as remote homology prediction and/or fold recognition.

In this paper we study and evaluate the feasibility of solving the $f\texttt{RMSD}$ estimation and reliability prediction problems for residue-pairs that are derived from optimal local sequence alignments. As a result, our evaluation focuses on the first two applications discussed in the previous paragraph (assessment of target-template alignment and identification of high-confidence alignment regions). However, the methods developed can also be used to address the other two applications as well.

4 Methods

We approach the problems of distinguishing reliable/unreliable residue-pairs and estimating their $f\texttt{RMSD}$ scores following a supervised machine learning framework and use support vector machines (SVM) [10, 36] to solve them.

Given a set of positive residue-pairs $\mathcal{A}^+$ (i.e., reliable) and a set of negative residue-pairs $\mathcal{A}^-$ (i.e., unreliable), the task of support vector classification is to learn a function $f(\pi)$ of the form

$$f(\pi) = \sum_{\pi_i \in \mathcal{A}^+} \lambda_i^+ \mathcal{K}(\pi, \pi_i) - \sum_{\pi_i \in \mathcal{A}^-} \lambda_i^- \mathcal{K}(\pi, \pi_i), \qquad (1)$$

where λ_i^+ and λ_i^- are non-negative weights that are computed during training by maximizing a quadratic objective function, and $\mathcal{K}(.,.)$ is the *kernel* function designed to capture the similarity between pairs of residue-pairs. Having learned the function $f(\pi)$, a new residue-pair π is predicted to be positive or negative depending on whether $f(\pi)$ is positive or negative. The value of $f(\pi)$ also signifies the tendency of π to be a member of the positive or negative class and can be

used to obtain a meaningful ranking of a set of the residue-pairs.

We use the error insensitive support vector regression ϵ-SVR [36, 34] for learning a function $f(\pi)$ to predict the $f\texttt{RMSD}(\pi)$ scores. Given a set of training instances $(\pi_i, f\texttt{RMSD}(\pi_i))$, the ϵ-SVR aims to learn a function of the form

$$f(\pi) = \sum_{\pi_i \in \Delta^+} \alpha_i^+ \mathcal{K}(\pi, \pi_i) - \sum_{\pi_i \in \Delta^-} \alpha_i^- \mathcal{K}(\pi, \pi_i), \qquad (2)$$

where Δ^+ contains the residue-pairs for which $f\texttt{RMSD}(\pi_i) - f(\pi_i) > \epsilon$, Δ^- contains the residue pairs for which $f\texttt{RMSD}(\pi_i) - f(\pi_i) < -\epsilon$, and α_i^+ and α_i^- are non-negative weights that are computed during training by maximizing a quadratic objective function. The objective of the maximization is to determine the flattest $f(\pi)$ in the feature space and minimize the estimation errors for instances in $\Delta^+ \cup \Delta^-$. Hence, instances that have an estimation error satisfying $|f(\pi_i) - f\texttt{RMSD}(\pi_i)| < \epsilon$ are neglected. The parameter ϵ controls the width of the regression deviation or tube.

In the current work we focused on several key considerations while setting up the classification and regression problems. In particular we explored different types of sequence information associated with the residue-pairs, developed efficient ways to encode this information to form fixed length feature vectors, and designed sensitive kernel functions to capture the similarity between pairs of residues in the feature spaces.

4.1 Sequence-based Information

For a given protein X, we encode the sequence information using profiles and predicted secondary structure.

4.1.1 Profile Information The profile of a protein X is derived by computing a multiple sequence alignment of X with a set of sequences $\{Y_1, \ldots, Y_m\}$ that have a statistically significant sequence similarity with X (i.e., they are sequence homologs).

The profile of a sequence X of length n is represented by two $n \times 20$ matrices, namely the position-specific scoring matrix $\mathcal{P}_X$ and the position-specific *frequency* matrix $\mathcal{F}_X$. Matrix $\mathcal{P}$ can be generated directly by running PSI-BLAST [2], whereas matrix $\mathcal{F}$ consists of the frequencies used by PSI-BLAST to derive $\mathcal{P}$. These frequencies, referred to as the *target frequencies* [18] consists of both the sequence-weighted observed frequencies (also referred to as *effective frequencies* [18]) and the BLOSUM62 [9] derived-pseudocounts [2]. Further, each row of the matrix $\mathcal{F}$ is normalized to one.

4.1.2 Predicted Secondary Structure Information For a sequence X of length n we predict the secondary structure and generate a position-specific secondary structure matrix $\mathcal{S}_X$ of length $n \times 3$. The (i, j) entry of this matrix represents the strength of the amino acid residue at position i to be in state j, where $j \in (0, 1, 2)$ corresponds to the three secondary structure elements: alpha helices (H), beta sheets (E), and coil regions (C).

4.2 Coding Schemes

The input to our prediction algorithms are a set of $w\texttt{mer}$-pairs associated with each residue-pair $\pi(x_i, y_j)$. The input feature space is derived using various combinations of the elements in the $\mathcal{P}$ and $\mathcal{S}$ matrices that are associated with the subsequences $w\texttt{mer}(x_i)$ and $w\texttt{mer}(y_j)$.

For the rest of this paper, we will use $\mathcal{P}_X(i-w \ldots i+w)$ to denote the $(2w+1)$ rows of matrix $\mathcal{P}_X$ corresponding to $w\texttt{mer}(x_i)$. A similar notation will be used for matrix $\mathcal{S}$.

4.2.1 Concatenation Coding Scheme For a given residue-pair $\pi(x_i, y_j)$, the feature-vector of the concatenation coding scheme is obtained by first linearizing the matrices $\mathcal{P}_X(i-w \ldots i+w)$ and $\mathcal{P}_Y(j-w \ldots j+w)$ and then concatenating the resulting vectors. This leads to feature-vectors of length $2 \times (2w+1) \times 20$. A similar representation is derived for matrix $\mathcal{S}$ leading to feature-vectors of length $2 \times (2w+1) \times 3$.

The concatenation coding scheme is order dependent as the representations for $\pi(x_i, y_j)$ and $\pi(y_j, x_i)$ are not equivalent. We call the feature representations obtained by the two concatenation orders as forward (*frwd*) and reverse (*rvsd*) representations. Note that we use the terms *forward* and *reverse* only for illustrative purposes as there is no way to assign a fixed ordering to the residues of a residue-pair, as this is the source of the problem in the first place.

We explored two different ways of addressing this order dependency. In the first approach, we trained up to ten models with random use of the forward and backward representation for the various instances. The final classification and regression results were determined by averaging the results produced by each of the ten different models. In the second approach, we built only one model based on the forward representation of the residue-pairs. However, during model application, we classified/regressed both the forward and reverse representations of a residue-pair and used the average of the SVM/ϵ-SVR outputs as the final classification/regression result. We denote this averaging method by *avg*.

4.2.2 Pairwise Coding Scheme For a given residue-pair $\pi(x_i, y_j)$, the pairwise coding scheme generates a feature-vector by linearizing the matrix formed by an element-wise product between $\mathcal{P}_X(i-w \ldots i+w)$ and $\mathcal{P}_Y(j-w \ldots j+w)$. The length of this vector is $(2w+1) \times 20$ and is order independent. If we denote the element-wise product operation by "$\otimes$", then the element-wise product matrix is given by

$$\mathcal{P}_X(-w+i \ldots w+i) \otimes \mathcal{P}_Y(-w+j \ldots w+j). \qquad (3)$$

A similar approach is used to obtain the pairwise coding scheme for matrix $\mathcal{S}$, leading to feature-vectors of length $(2w+1) \times 3$.

4.3 Kernel Functions

The general structure of the kernel function that we use for capturing the similarity between a pair of residue-pairs

$\pi(x_i, y_j)$ and $\pi'(x'_{i'}, y'_{j'})$ is given by

$$\mathcal{K}^{cs}(\pi, \pi') = \exp\left(1.0 + \frac{\mathcal{K}_1^{cs}(\pi, \pi')}{\sqrt{\mathcal{K}_1^{cs}(\pi, \pi)\,\mathcal{K}_1^{cs}(\pi', \pi')}}\right), \quad (4)$$

where $\mathcal{K}_1^{cs}(\pi, \pi')$ is given by

$$\mathcal{K}_1^{cs}(\pi, \pi') = \mathcal{K}_2^{cs}(\pi, \pi') + (\mathcal{K}_2^{cs}(\pi, \pi'))^2, \quad (5)$$

and $\mathcal{K}_2^{cs}(\pi, \pi')$ is a kernel function that depends on the choice of particular coding scheme (cs). For the concatenation coding scheme using matrix $\mathcal{P}$ (i.e., $cs = \mathcal{P}^{conc}$), $\mathcal{K}_2^{cs}(\pi, \pi')$ is given by

$$\mathcal{K}_2^{\mathcal{P}^{conc}}(\pi, \pi') = \sum_{k=-w}^{k=+w} \langle \mathcal{P}_X(i+k), \mathcal{P}_{X'}(i'+k) \rangle + \sum_{k=-w}^{k=+w} \langle \mathcal{P}_Y(j+k), \mathcal{P}_{Y'}(j'+k) \rangle. \quad (6)$$

For the pairwise coding scheme using matrix $\mathcal{P}$ (i.e., $cs = \mathcal{P}^{pair}$), $\mathcal{K}_2^{cs}(\pi, \pi')$ is given as

$$\mathcal{K}_2^{\mathcal{P}^{pair}}(\pi, \pi') = \sum_{k=-w}^{k=+w} \langle \mathcal{P}_X(i+k) \otimes \mathcal{P}_Y(j+k), \mathcal{P}_{X'}(i'+k) \otimes \mathcal{P}_{Y'}(j'+k) \rangle. \quad (7)$$

Similar kernel functions can be derived using matrix $\mathcal{S}$ for both the pairwise and the concatenation coding schemes. We will denote these coding schemes as $\mathcal{S}^{pair}$ and $\mathcal{S}^{conc}$, respectively. Since the overall structure of the kernel that we used (Equations 4 and 5) is that of a normalized second-order exponential function, we will refer to it as *nsoe*.

The second-order component of Equation 5 allows the *nsoe* kernel to capture pairwise dependencies among the residues used at various positions within each $w$mer, and we found that this leads to better results over the linear function. This observation is also supported by earlier research on secondary-structure prediction as well [14]. In addition, *nsoe*'s exponential function allows it to capture non-linear relationships within the data just like the kernels based on the Gaussian and radial basis function [36].

4.3.1 Fusion Kernels

We also developed a set of kernel functions that incorporate both profile and secondary structure information using an approach motivated by *fusion kernels* [16, 34]. Specifically, we constructed a new kernel function as the unweighted sum of the *nsoe* kernel function for the profile and secondary structure information. For example, the concatenation-based fusion kernel function is given by

$$\mathcal{K}^{(\mathcal{P}+\mathcal{S})^{conc}}(\pi, \pi') = \mathcal{K}^{\mathcal{P}^{conc}}(\pi, \pi') + \mathcal{K}^{\mathcal{S}^{conc}}(\pi, \pi'). \quad (8)$$

A similar kernel function can be defined for the pairwise coding scheme as well. We will denoted the pairwise-based fusion kernel by $\mathcal{K}^{(\mathcal{P}+\mathcal{S})^{pair}}(\pi, \pi')$. Note that since these fusion kernels are linear combinations of valid kernels, they are also admissible kernels.

5 Materials

5.1 Datasets

We evaluated the classification and regression performance of the various kernels on a set of protein pairs used in a previous study for learning a profile-to-profile scoring function [21]. These pairs of proteins were derived from the SCOP 1.57 database, classes a-e, with no two protein domains sharing greater than 75% sequence identity. The dataset is comprised of 473 protein pairs belonging to the same family, 433 pairs belonging to the same superfamily but not the same family, and 422 pairs belonging to the same fold but not the same superfamily. For each protein pair, we used the alignment produced by the Smith-Waterman [33] algorithm to generate the aligned residue-pairs that were used to train and test the various algorithms. These alignments were computed using the sensitive PICASSO [8, 18] profile-to-profile scoring function. For each aligned residue-pair $\pi(x_i, y_j)$, we computed its fRMSD(x_i, y_j) score by considering fragments of length seven (i.e., we optimally superimposed $v$frags with $v = 3$).

For the fRMSD estimation problem, we used the entire set of aligned residue-pairs and their corresponding fRMSD scores for training and testing the ϵ-SVR-based regression algorithms. For the reliability prediction problem, we used the aligned residue-pairs to construct two different classification datasets, that will be referred to as *easy* and *hard*. The positive class (i.e., reliable residue-pairs) for both datasets contains all residue-pairs whose fRMSD score is less than $0.75\mathring{A}$. However, the datasets differ on how the negative class (i.e., unreliable residue-pairs) is defined. For the hard problem, the negative class consists of all residue-pairs that are not part of the positive class (i.e., have an fRMSD score that is greater than or equal to $0.75\mathring{A}$), whereas for the easy problem, the negative class consists only of those residue-pairs whose fRMSD score is greater than $2.5\mathring{A}$. Thus, the easy dataset contains classes that are well-separated in terms of the fRMSD score of their residue-pairs and as such it represents a somewhat easier learning problem. Both these datasets are available at the supplementary website for this paper[1].

We perform a detailed analysis using different subsets of the datasets to train and test the performance of the models. Specifically, we train four models using (i) protein pairs sharing the same SCOP family, (ii) protein pairs sharing the same superfamily but not the family, (iii) protein pairs sharing the same fold but not the superfamily, and (iv) protein pairs from all the three levels. These four models are denoted by *fam*, *suf*, *fold*, and *all*. We also report performance numbers by splitting the test set in the aforementioned four levels. These subsets allow us to evaluate the performance of the schemes for different levels of sequence similarity.

5.2 Profile Generation

To generate the profile matrices $\mathcal{P}$ and $\mathcal{F}$, we ran PSI-BLAST, using the following parameters (`blastpgp -j 5 -e 0.01 -h 0.01`). The PSI-BLAST was performed against NCBI's nr database that was downloaded in November of 2004 and contained 2,171,938 sequences.

[1] http://bioinfo.cs.umn.edu/supplements/fRMSDPred/

5.3 Secondary Structure Prediction

We use the state-of-the-art secondary structure prediction server called YASSPP [14] (default parameters) to generate the $\mathcal{S}$ matrix. The values of the $\mathcal{S}$ matrix are the output of the three one-versus-rest SVM classifiers trained for each of the secondary structure elements.

5.4 Evaluation Methodology

We use a five-fold cross-validation framework to evaluate the performance of the various classifiers and regression models. To prevent unwanted biases, we restrict all residue-pairs involving a particular protein to belong solely in the training or the testing dataset.

We measure the quality of the methods using the standard receiver operating characteristic (ROC) scores and the ROC_5 scores averaged across every protein pair. The ROC score is the normalized area under the curve that plots the true positives against the false positives for different thresholds for classification [7]. The ROC_n score is the area under the ROC curve up to the first n false positives. We compute the ROC and ROC_5 numbers for every protein pair and report the average results across all the pairs and cross-validation steps. We selected to report ROC_5 scores because each individual ROC-based evaluation is performed on a per protein-pair basis, which, on average, involves one to two hundred residue-pairs.

The regression performance is assessed by computing the standard Pearson correlation coefficient (CC) between the predicted and observed $f\texttt{RMSD}$ values for every protein pair. The results reported are averaged across the different pairs and cross-validation steps.

5.5 Profile-to-Profile Scoring schemes

To assess the effectiveness of our supervised learning algorithms we compare their performance against that obtained by using two profile-to-profile scoring schemes to solve the same problems. Specifically, we use the profile-to-profile scoring schemes to compute the similarity between the aligned residue-pairs summed over the length of their $w\texttt{mers}$. To assess how well these scores correlated with the $f\texttt{RMSD}$ score of each residue-pair we compute their correlation coefficients. Note that since residue-pairs with high-similarity score are expected to have low $f\texttt{RMSD}$ scores, good values for these correlation coefficients will be close to -1. Similarly, for the reliability prediction problem, we sort the residue-pairs in decreasing similarity score order and assess the performance by computing ROC and ROC_5 scores.

The two profile-to-profile scoring schemes that we used are based on the dot-product and the PICASSO score, both of which are used extensively and shown to produce good results [18, 39, 17]. The dot-product similarity score is defined both for the profile- as well as the secondary-structure-based information, whereas the PICASSO score is defined only for the profile-based information. The profile-based dot-product similarity score between residues x_i and y_j is given by $\langle \mathcal{P}_X(i), \mathcal{P}_Y(j) \rangle$. Similarly, the secondary-structure-based dot-product similarity score is given by $\langle \mathcal{S}_X(i), \mathcal{S}_Y(j) \rangle$. The PICASSO similarity score [8, 18] between residues x_i and y_j uses both the $\mathcal{P}$ and $\mathcal{F}$ matrices and is given by $\langle \mathcal{F}_X(i)\,\mathcal{P}_Y(j) + \mathcal{F}_Y(j)\,\mathcal{P}_X(i) \rangle$. We will use $\mathcal{P}_{dotp}$, $\mathcal{S}_{dotp}$, and $\mathcal{PF}_{pic}$ to denote these three similarity scores, respectively.

5.6 Support Vector Machines

The classification and regression is done using the publicly available support vector machine tool SVM^{light} [29] that implements an efficient soft margin optimization algorithm.

The performance of SVM and ϵ-SVR depends on the parameter that controls the trade-off between the margin and the misclassification cost ("C" parameter). In addition, the performance of ϵ-SVR also depends on the value of the deviation parameter ϵ. We performed a limited number of experiments to determine good values for these parameters. These experiments showed that $C = 0.1$ and $\epsilon = 0.1$ achieved consistently good performance and was the value used for all the reported results.

6 Results

We have performed a comprehensive study evaluating the classification and regression performance of the various information sources, coding schemes, and kernel functions (Section 4) and compare it against the performance achieved by the profile-to-profile scoring schemes (Section 5.5).

We performed a number of experiments using different length $w\texttt{mers}$ for both the SVM/ϵ-SVR- and profile-to-profile-based schemes. These experiments showed that the supervised learning schemes achieved the best results when $5 \leq w \leq 7$, whereas in the case of the profile-to-profile scoring schemes, the best performing value of w was dependent on the particular scoring scheme. For these reasons, for all the SVM/ϵ-SVR-based schemes we only report results for $w = 6$, whereas for the profile-to-profile schemes we report results for the values of w that achieved the best performance.

6.1 Order Dependency in the Concatenation Coding Scheme

Section 4.2.1 described two different schemes for addressing the order-dependency of the concatenation coding scheme. Our experiments with these approaches showed that both achieved comparable results. For this reason and due to space constraints in this section we only present results for the second approach (i.e., averaging the SVM/ϵ-SVR prediction values of the forward and reverse representations). These results are shown in Table 1, which shows the classification and regression performance achieved by the concatenation-based fusion kernel for the two representations and their average.

These results show that there exists a difference in the performance achieved by the forward and reverse representations. Depending on the protein set used to train and/or test the model, these differences can be non-trivial. For example, for models trained on the *fold* and *all* protein sets, the performance achieved by the reverse representation is considerably higher than that achieved by the forward representation. However, these results also show that by averaging

Table 1: Comparing the classification and regression performance of the various concatenation based kernels due to order dependency.

	Reliability Prediction				EST
	EASY		HARD		
Scheme	ROC_5	ROC	ROC_5	ROC	CC
$(\mathcal{P}+\mathcal{S})^{conc}$ *-fam* (*frwd*)	0.802	0.937	0.666	0.903	0.693
$(\mathcal{P}+\mathcal{S})^{conc}$ *-fam* (*rvsd*)	0.803	0.937	0.664	0.902	0.693
$(\mathcal{P}+\mathcal{S})^{conc}$ *-fam* (*avg*)	**0.817**	0.941	**0.673**	0.906	**0.700**
$(\mathcal{P}+\mathcal{S})^{conc}$ *-suf* (*frwd*)	0.822	0.938	0.653	0.898	0.687
$(\mathcal{P}+\mathcal{S})^{conc}$ *-suf* (*rvsd*)	0.821	0.938	0.651	0.899	0.688
$(\mathcal{P}+\mathcal{S})^{conc}$ *-suf* (*avg*)	**0.827**	0.940	**0.659**	0.902	**0.694**
$(\mathcal{P}+\mathcal{S})^{conc}$ *-fold* (*frwd*)	0.785	0.918	0.618	0.872	0.660
$(\mathcal{P}+\mathcal{S})^{conc}$ *-fold* (*rvsd*)	**0.800**	0.922	**0.638**	0.881	0.663
$(\mathcal{P}+\mathcal{S})^{conc}$ *-fold* (*avg*)	0.796	0.922	0.637	0.882	**0.667**
$(\mathcal{P}+\mathcal{S})^{conc}$ *-all* (*frwd*)	0.839	0.948	0.680	0.909	0.717
$(\mathcal{P}+\mathcal{S})^{conc}$ *-all* (*rvsd*)	0.853	0.950	0.692	0.913	0.721
$(\mathcal{P}+\mathcal{S})^{conc}$ *-all* (*avg*)	**0.853**	0.952	**0.693**	0.913	**0.725**

The test set consisted of proteins from the *all* set, whereas the training set uses either the *all*, *fam*, *suf*, and *fold* sets. The *frwd* and *rvsd* notations indicate concatenation orders of the two $w$mers, whereas *avg* denotes the scheme which uses the average output of both the results. EST denotes the fRMSD estimation results using regression. The numbers in bold show the best performing schemes for each of the sub-tables.

Table 2: Comparing the performance of the rbf and nsoe kernel functions.

	Reliability Prediction				EST
	EASY		HARD		
Scheme	ROC_5	ROC	ROC_5	ROC	CC
$\mathcal{P}^{conc}$*-all* (*rbf*)	0.728	0.910	0.572	0.865	0.537
$\mathcal{P}^{conc}$*-all* (*nsoe*)	**0.750**	0.918	**0.598**	0.875	**0.566**
$\mathcal{P}^{pair}$*-all* (*rbf*)	0.708	0.900	0.550	0.854	0.528
$\mathcal{P}^{pair}$*-all* (*nsoe*)	**0.723**	0.905	**0.559**	0.856	**0.534**

The test and training set consisted of proteins from the *all* set. EST denotes the fRMSD estimation results using regression. The numbers in bold show the best performing schemes for each of the sub-tables.

the predictions of these two representations, we are able to achieve the best results (or close to). In many cases, the averaging scheme achieves up to 1% improvement over either the forward or reverse representations for both the classification as well as regression problem. For this reason, throughout the rest of this study we only report the results obtained using the averaging scheme for the concatenation-based coding schemes.

6.2 RBF versus NSOE Kernel Functions

Table 2 compares the classification and regression performance achieved by the standard *rbf* kernel against that achieved by the normalized second-order exponential kernel (*nsoe*) described in Section 4.3. These results are reported only for the concatenation and pairwise coding schemes that use profile information. The *rbf* results were obtained after normalizing the feature-vectors to unit length, as it produced substantially better results over the unnormalized representation.

These results show that the performance achieved by the *nsoe* kernel is consistently 3% to 5% better than that achieved by the *rbf* kernel for both the classification and regression problems. The key difference between the two kernels is that in the *nsoe* kernel the even-ordered terms are weighted higher in the expansion of the infinite exponential series than the *rbf* kernel. As discussed in Section 4.3, this allows the *nsoe* kernel function to better capture the pairwise dependencies that exists at different positions of each $w$mer.

6.3 Input Information and Coding Schemes

Table 3 compares how the features derived from the profiles and the predicted secondary structure impact the performance achieved for the reliability prediction problem. The table presents results for the SVM-based schemes using the concatenation and pairwise coding schemes as well as results obtained by the dot-product-based profile-to-profile scoring scheme (see the discussion in Section 5.5 for a discussion on how these scoring schemes were used to solve the reliability prediction problem).

Analyzing these results across the different SCOP-derived test sets, we can see that protein profiles lead to better performance for the family-derived set, whereas secondary structure information does better for the superfamily- and fold-derived sets. The performance improvements achieved by the secondary-structure-based schemes are usually much greater than the improvements achieved by the profile-based scheme. Moreover, the relative performance gap between secondary-structure- and profile-based schemes increases as we move from the superfamily- to the fold-derived set. This holds for both the easy and hard datasets and for both the kernel-based methods and the profile-to-profile-based scoring scheme. These results show that profiles are more important for protein-pairs that are similar (as it is the case in the family-derived set), whereas secondary-structure information becomes increasingly more important as the sequence similarity between the protein-pairs decreases (as it is the case in the superfamily- and fold-derived sets).

Analyzing the performance achieved by the different coding schemes, we can see that concatenation performs uniformly better than pairwise. As measured by ROC_5, the concatenation scheme achieves 4% to 15% better performance than the corresponding pairwise-based schemes. However, both schemes perform considerably better than the profile-to-profile-based scheme. These performance advantages range from 11% to 30% (as measured by ROC_5).

6.4 Fusion Kernels

6.4.1 Reliability Prediction Problem Table 4 shows the performance achieved by the fusion kernels on solving the reliability prediction problem for both the easy and hard datasets. For comparison purposes, this table also shows the best results that were obtained by using the profile-to-profile-based schemes to solve the reliability prediction problem. Specifically, we present dot-product-based results that score each $w$mer as the sum of its profile and secondary-structure information ($(\mathcal{P}+\mathcal{S})_{dotp}$) and results that score each $w$mer as the sum of its PICASSO score and a secondary-structure-

Table 3: Classification performance of the individual kernels for both the easy and hard datasets.

	EASY						HARD					
	fam		*suf*		*fold*		*fam*		*suf*		*fold*	
Scheme	ROC_5	ROC	ROC_5	ROC	ROC_5	ROC	ROC_5	ROC	ROC_5	ROC	ROC_5	ROC
$\mathcal{P}_{dotp}$ (6)	**0.673**	0.826	0.496	0.803	0.341	0.717	**0.470**	0.753	0.315	0.698	0.236	0.646
$\mathcal{S}_{dotp}$ (3)	0.642	0.786	**0.680**	0.884	**0.706**	0.901	0.466	0.771	**0.503**	0.856	**0.567**	0.885
$\mathcal{P}^{conc}$-*all*	**0.817**	0.919	0.716	0.917	0.712	0.918	**0.621**	0.867	0.574	0.880	0.590	0.882
$\mathcal{S}^{conc}$-*all*	0.790	0.908	**0.794**	0.939	**0.823**	0.951	0.615	0.865	**0.631**	0.913	**0.695**	0.923
$\mathcal{P}^{pair}$-*all*	**0.784**	0.902	**0.699**	0.909	0.679	0.905	**0.588**	0.849	0.509	0.853	0.572	0.868
$\mathcal{S}^{pair}$-*all*	0.676	0.837	0.690	0.909	**0.727**	0.922	0.486	0.803	**0.548**	0.880	**0.636**	0.895

The test set consisted of proteins from the *fam*, *suf*, and *fold* sets, whereas the training set used the *all* set. The numbers in parentheses for the profile-to-profile scoring schemes indicate the value of w for the $w$mers that were used. The numbers in bold show the best performing schemes for each of the sub-tables.

Table 4: Classification performance of the fusion kernels for the easy and hard datasets.

	EASY								HARD							
	all		*fam*		*suf*		*fold*		*all*		*fam*		*suf*		*fold*	
Scheme	ROC_5	ROC	ROC_5	ROC	ROC_5	ROC	ROC_5	ROC	ROC_5	ROC	ROC_5	ROC	ROC_5	ROC	ROC_5	ROC
$(\mathcal{P}+\mathcal{S})_{dotp}(6)$	0.523	0.794	0.679	0.831	0.511	0.814	0.359	0.733	0.365	0.716	0.474	0.758	0.328	0.710	0.249	0.663
$\mathcal{PF}_{pic}+\mathcal{S}_{dotp}(2)$	**0.719**	0.891	**0.733**	0.865	**0.720**	0.911	**0.701**	0.901	**0.526**	0.850	**0.535**	0.820	**0.498**	0.864	**0.543**	0.878
$(\mathcal{P}+\mathcal{S})^{conc}$ -*fam*	0.817	0.941	0.829	0.929	0.811	0.948	0.808	0.949	0.673	0.906	0.652	0.879	0.662	0.921	0.714	0.927
$(\mathcal{P}+\mathcal{S})^{conc}$ -*suf*	0.827	0.940	0.820	0.918	0.821	0.948	0.841	0.957	0.659	0.902	0.610	0.866	0.676	0.925	0.711	0.929
$(\mathcal{P}+\mathcal{S})^{conc}$ -*fold*	0.796	0.922	0.751	0.874	0.778	0.931	0.863	0.967	0.637	0.882	0.557	0.822	0.635	0.903	**0.753**	0.944
$(\mathcal{P}+\mathcal{S})^{conc}$ -*all*	**0.853**	0.952	**0.846**	0.936	**0.841**	0.956	**0.873**	0.967	**0.693**	0.913	**0.665**	0.886	**0.679**	0.926	0.747	0.939
$(\mathcal{P}+\mathcal{S})^{pair}$ -*fam*	0.783	0.925	0.797	0.909	0.786	0.939	0.762	0.930	0.640	0.888	0.621	0.863	0.627	0.899	0.681	0.911
$(\mathcal{P}+\mathcal{S})^{pair}$ -*suf*	0.810	0.932	0.805	0.907	0.818	0.945	0.808	0.947	0.652	0.890	0.619	0.859	0.653	0.904	0.698	0.919
$(\mathcal{P}+\mathcal{S})^{pair}$ -*fold*	0.805	0.923	0.765	0.879	0.799	0.937	0.855	0.959	0.644	0.882	0.576	0.837	0.636	0.894	0.751	0.936
$(\mathcal{P}+\mathcal{S})^{pair}$ -*all*	0.832	0.942	0.823	0.920	0.825	0.949	0.850	0.958	0.668	0.897	0.634	0.867	0.650	0.907	0.734	0.930

The test and training set consisted of proteins from the *all*, *fam*, *suf*, and *fold* sets. The numbers in parentheses for the profile-to-profile scoring schemes indicate the value of w for the $w$mers that were used. The numbers in bold show the best performing schemes for the kernel-based and profile-to-profile scoring based schemes. The underlined results show the cases where the pairwise coding scheme performs better than the concatenation coding scheme.

based dot-product score $(\mathcal{PF}_{pic} + \mathcal{S}_{dotp})$.

From these results we can see that the SVM-based schemes, regardless of their coding schemes, consistently outperform the profile-to-profile scoring schemes. In particular, comparing the best results obtained by the concatenation scheme against those obtained by the $\mathcal{PF}_{pic} + \mathcal{S}_{dotp}$ scheme (i.e., entries in bold), we see that the former achieves 18% to 24% higher ROC_5 scores for the easy dataset. Moreover, the performance advantage becomes greater for the hard dataset and ranges between 31% to 36%.

Comparing the performance achieved by the fusion kernels with that achieved by the *nsoe* kernels (Table 3) we can see that by combing both profile and secondary structure information we can achieve an ROC_5 improvement between 3.5% and 10.8%. These performance improvements are consistent across the different test sets (*fam*, *suf*, and *fold*) and datasets (hard and easy).

Comparing the performance achieved by the models trained on different protein subsets, we can see that the best performance is generally achieved by models trained on protein pairs from all three levels of the SCOP hierarchy (i.e., trained using the *all* set). However, these results also show an interesting trend that involves the set of fold-derived protein-pairs. For this set, the best (or close to) classification performance is achieved by models trained on fold-derived protein-pairs. This holds for both the concatenation and pairwise coding schemes and the easy and hard datasets. These results indicate that training a model using residue-pairs with high-to-moderate sequence similarity (i.e., as it is the case with the *fam*- and *suf*-derived sets) does not perform very well for predicting reliable residue-pairs that have low or no sequence similarity (as it is the case with the *fold*-derived set).

Finally, as it was the case with the *nsoe* kernels, the concatenation coding schemes tend to outperform the pairwise schemes for the fusion kernels as well. However, the advantage of the concatenation coding scheme is not uniform and there are certain training and test set combinations for which the pairwise scheme does better. These cases correspond to the underlined entries in Table 4.

6.4.2 fRMSD Estimation Problem

Table 5 shows the performance achieved by ϵ-SVR for solving the fRMSD estimation problem as measured by the correlation coefficient between the observed and predicted fRMSD values. We report results for the fusion kernels and the $\mathcal{PF}_{pic} + \mathcal{S}_{dotp}$ profile-to-profile scoring scheme. Note that as discussed in Section 5.5, the scores computed by $\mathcal{PF}_{pic} + \mathcal{S}_{dotp}$ should be negatively correlated with the fRMSD; thus, negative correlations represent good estimations.

From these results we can see that as it was the case with the reliability prediction problem, the ϵ-SVR-based methods consistently outperform the profile-to-profile scoring scheme

Table 5: Regression Performance of the fusion kernels on the hard dataset.

Scheme	*all*	*fam*	*suf*	*fold*
$\mathcal{PF}_{pic} + \mathcal{S}_{dotp}$ (3)	-0.590	-0.550	-0.611	-0.625
$(\mathcal{P}+\mathcal{S})^{conc}$ *-fam*	0.700	0.662	0.720	0.736
$(\mathcal{P}+\mathcal{S})^{conc}$ *-suf*	0.694	0.612	0.739	0.764
$(\mathcal{P}+\mathcal{S})^{conc}$ *-fold*	0.667	0.557	0.719	**0.770**
$(\mathcal{P}+\mathcal{S})^{conc}$ *-all*	**0.725**	**0.681**	**0.744**	0.768
$(\mathcal{P}+\mathcal{S})^{pair}$ *-fam*	0.676	0.639	0.695	0.708
$(\mathcal{P}+\mathcal{S})^{pair}$ *-suf*	0.672	0.610	0.705	0.727
$(\mathcal{P}+\mathcal{S})^{pair}$ *-fold*	0.676	0.639	0.695	0.708
$(\mathcal{P}+\mathcal{S})^{pair}$ *-all*	0.694	0.645	0.712	0.746

The test and training set consisted of proteins from the *all*, *fam*, *suf*, and *fold* sets. The number in parentheses for the profile-to-profile scoring scheme indicates the value of w for the $w$mer that was used. Good correlation coefficient values will be negative for the profile-to-profile scoring scheme and positive for the kernel-based schemes. The numbers in bold show the best performing schemes. The underlined results show the cases where the pairwise coding scheme performs better than the concatenation coding scheme.

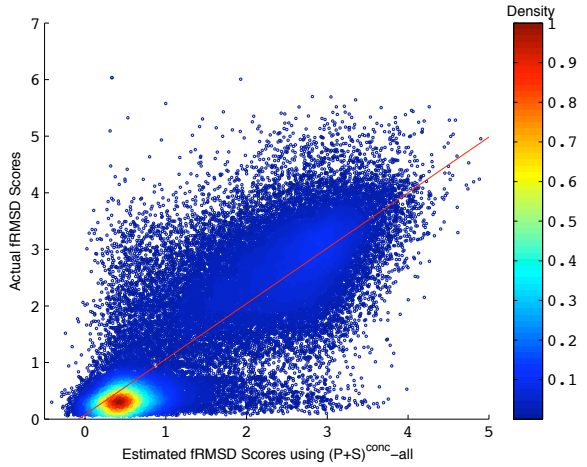

Figure 1: Scatter plot for test protein-pairs at all levels between estimated and actual fRMSD scores. The color coding represents the approximate density of points plotted in a fixed normalized area.

across the different combinations of training and testing sets. The $(\mathcal{P}+\mathcal{S})^{conc}$ models achieve an improvement over $\mathcal{PF}_{pic} + \mathcal{S}_{dotp}$ that ranges from 21% to 23.2%. The performance difference between the two schemes can also be seen in Figures 1 and 2 that plots the actual fRMSD scores against the estimated fRMSD scores of $(\mathcal{P}+\mathcal{S})^{conc}$*-all* and the $\mathcal{PF}_{pic} + \mathcal{S}_{dotp}$ similarity scores, respectively. Comparing the two figures we can see that the fRMSD estimations produced by the ϵ-SVR-based scheme are significantly better correlated with those produced by $\mathcal{PF}_{pic} + \mathcal{S}_{dotp}$.

Finally, in agreement with the earlier results, the concatenation coding scheme performs better than the pairwise scheme. The only exceptions are the models trained on the *fold*-derived set, for which the pairwise scheme does better when tested on the *all*- and *fam*-derived sets (underlined entries in Table 5).

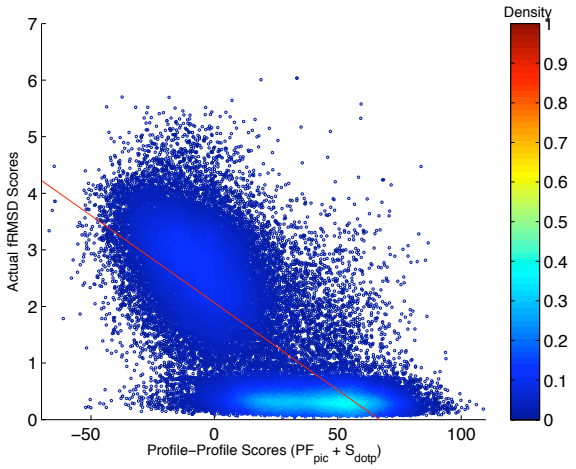

Figure 2: Scatter plot for test protein-pairs at all levels between profile-to-profile scores and actual fRMSD scores. The color coding represents the approximate density of points plotted in a fixed normalized area.

7 Related Research

The problem of determining the reliability of residue-pairs has been visited before in several different settings. ProfNet [21, 20] uses artificial neural networks to learn a scoring function to align a pair of protein sequences. In essence, ProfNet aims to differentiate related and unrelated residue-pairs and also estimate the RMSD score between these residue-pairs using profile information. Protein pairs are aligned using STRUCTAL [6], residue-pairs within $3\mathring{A}$ apart are considered to be related, and unrelated residue-pairs are selected randomly from protein pairs known to be in different folds. A major difference between our methods and ProfNet is in the definition of reliable/unreliable residue-pairs and on how the RMSD score between residue-pairs is measured. As discussed in Section 2, we measure the structural similarity of two residues (fRMSD) by looking at how well their vfrags structurally align with each other. However, ProfNet only considers the proximity of two residues within the context of their global structural alignment. As such, two residues can have a very low RMSD and still correspond to fragments whose structure is substantially different. This fundamental difference makes direct comparisons between the results impossible. The other major differences lie in the development of order independent coding schemes and the use of information from a set of neighboring residues by using a $w$mer size greater than zero.

The task of aligning a pair of sequences has also been casted as a problem of learning parameters (gap opening, gap extension, and position independent substitution matrix) within the framework of discriminatory learning [11, 40] and setting up optimization parameters for an inverse learning problem [35]. Recently, pair conditional random fields were also used to learn a probabilistic model for estimating the alignment parameters (i.e., gap and substitution costs) [4].

8 Conclusion and Future Work

In this paper we defined the $f\texttt{RMSD}$ estimation and the reliability prediction problems to capture the local structural similarity using only sequence-derived information. We developed a machine-learning approach for solving these problems by using a second-order exponential kernel function to encode profile and predicted secondary structure information into a kernel fusion framework. Our results showed that the $f\texttt{RMSD}$ values of aligned residue-pairs can be predicted at a good level of accuracy. We believe that this lays the foundation for using estimated $f\texttt{RMSD}$ values to evaluate the quality of target-template alignments and refine them.

9 Acknowledgment

We would like to express our deepest thanks to Professor Arne Elofsson and Dr. Tomas Ohlson for helping us with datasets for the study. This work was supported by NSF EIA-9986042, ACI-0133464, IIS-0431135, NIH RLM008713A, the Army High Performance Computing Research Center contract number DAAD19-01-2-0014, and by the Digital Technology Center at the University of Minnesota.

References

[1] S. F. Altschul, W. Gish, E. W. Miller, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[2] S. F. Altschul, L. T. Madden, A. A. Schffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–402, 1997.

[3] Helen M. Berman, T. N. Bhat, Philip E. Bourne, Zukang Feng, Gary Gilliland Helge Weissig, and John Westbrook. The Protein Data Bank and the challenge of structural genomics. *Nature Structural Biology*, 7:957–959, November 2000.

[4] C. B. Do, S. S. Gross, and S. Batzoglou. Contralign: Discriminative training for protein sequence alignment. *In Proceedings of the Tenth Annual International Conference on Computational Molecular Biology (RECOMB)*, 2006.

[5] R. Edgar and K. Sjolander. A comparison of scoring functions for protein sequence profile alignment. *BIOINFORMATICS*, 20(8):1301–1308, 2004.

[6] M. Gerstein and M. Levitt. Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins. *Protein Science*, 7:445–456, 1998.

[7] M. Gribskov and N. Robinson. Use of receiver operating characteristic (roc) analysis to evaluate sequence matching. *Computational Chemistry*, 20:25–33, 1996.

[8] A. Heger and L. Holm. Picasso:generating a covering set of protein family profiles. *Bioinformatics*, 17(3):272–279, 2001.

[9] S. Henikoff and J. G. Henikoff. Amino acid subsitution matrices from protein blocks. *PNAS*, 89:10915–10919, 1992.

[10] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proc. of the European Conference on Machine Learning*, 1998.

[11] T. Joachims, T. Galor, and R. Elber. Learning to align sequences: A maximum-margin approach. *New Algorithms for Macromolecular Simulation*, 49, 2005.

[12] D. T. Jones. Genthreader: an efficient and reliable protein fold recognition method for genomic sequences. *Journal of Molecular Biology*, 287:797–815, 1999.

[13] D. T. Jones, W. R. Taylor, and J. M. Thorton. A new approach to protein fold recognition. *Nature*, 358:86–89, 1992.

[14] George Karypis. Yasspp: Better kernels and coding schemes lead to improvements in svm-based secondary structure prediction. *Proteins: Structure, Function and Bioinformatics*, 64(3):575–586, 2006.

[15] A. S. Konagurthu, J. C. Whisstock, P. J. Stuckey, and A. M. Lesk. Mustang: a multiple structural alignment algorithm. *Proteins: Structure, Function, and Bioinformatics*, 64(3):559–574, 2006.

[16] G. R. G. Lanckriet, T. D. Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004.

[17] M. Marti-Renom, M. Madhusudhan, and A. Sali. Alignment of protein sequences by their profiles. *Protein Science*, 13:1071–1087, 2004.

[18] D. Mittelman, R. Sadreyev, and N. Grishin. Probabilistic scoring measures for profile-profile comparison yield more accurate short seed alignments. *Bioinformatics*, 19(12):1531–1539, 2003.

[19] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453, 1970.

[20] T. Ohlson, V. Aggarwal, A. Elofsson, and R. Maccallum. Improved alignment quality by combining evolutionary information, predicted secondary structure and self-organizing maps. *BMC Bioinformatics*, 1(357), 2006.

[21] T. Ohlson and A. Elofsson. Profnet, a method to derive profile-profile alignment scoring functions that improves the alignments of distantly related proteins. *BMC Bioinformatics*, 6(253), 2005.

[22] William R. Pearson and David J. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85:2444–2448, 1988.

[23] J. Pillardy, C. Czaplewski, A. Liwo, J. Lee, D. R. Ripoll, R. Kazmierkiewicz, S. Oldziej, W. J. Wedemeyer, K. D. Gibson, Y. A. Arnautova, J. Saunders, Y. J. Ye, and H. A. Scheraga. Recent improvements in prediction of protein structure by global optimization of a potential energy function. *PNAS USA*, 98(5):2329–2333, 2001.

[24] J. Qiu and R. Elber. Ssaln: An alignment algorithm using structure-dependent substitution matrices and gap penalties learned from structurally aligned protein pairs. *Proteins: Structure, Function, and Bioinformatics*, 62(4):881–891, 2006.

[25] H. Rangwala and G. Karypis. Incremental window-based protein sequence alignment algorithms. *Bioinformatics*, 23(2):e17–23, 2007.

[26] C. A. Rohl, C. E. M. Strauss, K. M. S. Misura, and D. Baker. Protein structure prediction using rosetta. *Methods in Enzymology*, 383:66–93, 2004.

[27] R. Sanchez and A. Sali. Advances in comparative protein-structure modelling. *Current Opinion in Structural Biology*, 7(2):206–214, 1997.

[28] T. Schwede, J. Kopp, N. Guex, and M. C. Peltsch. Swiss-model: An automated protein homology-modeling server. *Nucleic Acids Research*, 31(13):3381–3385, 2003.

[29] B. Schlkopf, C. Burges, and A. Smola, editors. *Making large-Scale SVM Learning Practical*. Advances in Kernel Methods - Support Vector Learning. MIT Press, 1999.

[30] I. Shindyalov and P. E. Bourne. Protein structure alignment by incremental combinatorial extension (ce) of the optimal path. *Protein Engineering*, 11:739–747, 1998.

[31] K. T. Simons, C. Strauss, and D. Baker. Prospects for ab initio protein structural genomics. *Journal of Molecular Biology*, 306(5):1191–1199, 2001.

[32] J. Skolnick and D. Kihara. Defrosting the frozen approximation: Prospector–a new approach to threading. *Proteins: Structure, Function and Genetics*, 42(3):319–331, 2001.

[33] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[34] A. Smola and B. Scholkopf. A tutorial on support vector regression. *NeuroCOLT2*, NC2-TR-1998-030, 1998.

[35] F. Sun, D. Fernandez-Baca, and W. Yu. Inverse parametric sequence alignment. *Proceedings of the International Computing and Combinatorics Conference (COCOON)*, 2002.

[36] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.

[37] C. Venclovas. Comparative modeling in casp5: Progress is evident, but alignment errors remain a significant hindrance. *Proteins: Structure, Function, and Genetics*, 53:380–388, 2003.

[38] C. Venclovas and M. Margelevicius. Comparative modeling in casp6 using consensus approach to template selection, sequence-structure alignment, and structure assessment. *Proteins: Structure, Function, and Bioinformatics*, 7:99–105, 2005.

[39] G. Wang and R. L. Dunbrack JR. Scoring profile-to-profile sequence alignments. *Protein Science*, 13:1612–1626, 2004.

[40] C. Yu, T. Joachims, R. Elber, and J. Pillardy. Support vector training of protein alignment models. *To appear in Proceeding of the Eleventh International Conference on Research in Computational Molecular Biology (RECOMB)*, 2007.

[41] Y. Zhang, A. J. Arakaki, and J. Skolnick. Tasser: an automated method for the prediction of protein tertiary structures in casp6. *Proteins: Structure, Function, and Bioinformatics*, 7:91–98, 2005.

[42] Y. Zhang and J. Skolnick. The protein structure prediction problem could be solved using the current pdb library. *PNAS USA*, 1024(4):1029–1034, 2005.