Fold Recognition by Predicted Alignment Accuracy

Jinbo Xu

Abstract—One of the key components in protein structure prediction by protein threading technique is to choose the best overall template for a given target sequence after all the optimal sequence-template alignments are generated. The chosen template should have the best alignment with the target sequence since the three-dimensional structure of the target sequence is built on the sequence-template alignment. The traditional method for template selection is called Z-score, which uses a statistical test to rank all the sequence-template alignments and then chooses the first-ranked template for the sequence. However, the calculation of Z-score is time-consuming and not suitable for genome-scale structure prediction. Z-scores are also hard to interpret when the threading scoring function is the weighted sum of several energy items of different physical meanings. This paper presents a Support Vector Machine (SVM) regression approach to directly predict the alignment accuracy of a sequence-template alignment, which is used to rank all the templates for a specific target sequence. Experimental results on a large-scale benchmark demonstrate that SVM regression performs much better than the composition-corrected Z-score method. SVM regression also runs much faster than the Z-score method.

Index Terms—Protein structure prediction, protein threading, protein fold recognition, SVM regression.

1 Introduction

Protein structure prediction by protein threading technique has demonstrated a great success in recent CASPs (Critical Assessment of Structure Prediction) [1], [2], [3]. Given a target sequence, protein threading makes a structure prediction for this target by finding the optimal alignment between this target sequence and each of the available protein structures (also called templates) in a template library, a subset of Protein Data Bank (PDB), and then building the three-dimensional structure based on the best sequence-template alignment. Protein structure prediction by protein threading technique consists of the following five major steps:

- Step 1: Construct a protein structure template library from PDB by excluding highly similar proteins. Usually, a template library contains only several protein structures with a similar fold.
- Step 2: Design a scoring function to measure the quality of sequence-template alignment.
- Step 3: Design an efficient algorithm to optimize the scoring function, which leads to the optimal sequence-template alignment for each sequence-template pair. Align the target sequence to each of all the templates in the library.
- Step 4: Choose the template with the best alignment to the target sequence.
- Step 5: Build the three-dimensional structure for the target sequence based on the alignment between the sequence and the chosen template.

Construction of a structure template library can be easily done using a structure clustering tool. Step 5 can also be readily done by some homology modeling tools such as MODELLER [4] as long as we have a high-quality sequence-template alignment. Design of scoring function and optimal sequence-template alignment algorithms have been researched extensively [5], [6], [7], [8], [9], [10], [11], [12]. However, how to choose the best template (or the sequence-template alignment with the best alignment accuracy) based on all the sequence-structure alignments does not gain enough attention, although this step is also critical to the success of protein threading. The higher accuracy the sequence-template alignment has, the better three-dimensional structure we can generate for the target sequence. In this paper, we will develop a SVM regression-based method to attack this problem. The task of choosing the best template for a given target sequence is also called fold recognition in the protein structure prediction community.

Fold recognition requires a criterion to identify the best template for one target sequence. Please notice here that simply choosing a template with a similar fold as the target sequence is not enough for structure prediction. To build the three-dimensional structure of a target sequence, a high-quality sequence-template alignment is indispensable. A high-quality sequence-template alignment cannot be obtained easily unless the structures of both proteins are available. The sequence-template alignment score cannot be directly used to rank the templates due to the bias introduced by the residue composition and the number of alternative sequence-template alignments [13]. So far, there are two strategies used by the structure prediction community for fold recognition: recognition based on Z-scores [13], and recognition by machine learning methods [9], [10]. Most of the current prediction programs use the traditional Z-score to recognize the best-fit templates, whereas several programs such as GenTHREADER [9] and PROSPECT-I [14][1] use

• *The author is with the School of Computer Science, University of Waterloo, Waterloo, Ontario Canada N2L 3G1. E-mail: j3xu@uwaterloo.ca.*

Manuscript received 8 Aug. 2004; revised 7 Dec. 2004; accepted 13 Dec. 2004; published online 2 June 2005.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBBSI-0084-0804.

1. The draft version of PROSPECT-II [6] also proposed an SVM classification method, but the final version did not include it.

a neural network to rank the templates. The neural network method treats the template selection problem as a classification problem. We will argue later that treating the template selection problem as a classification problem is not good enough for three-dimensional structure prediction. Z-score was proposed to cancel out the bias caused by sequence residue composition and by the number of alternative sequence-template alignments. To cancel out the bias caused by sequence residue composition, Bryant and Altschul [13] proposed the following procedures:

- Fix the optimal alignment positions between the target sequence and the template.
- Shuffle the aligned sequence residues randomly.
- Calculate the alignment scores based on the fixed alignment.
- Repeat the above three steps N times (N is on the order of several thousands).

The Z-score is the alignment score in standard deviation units relative to the mean of all the alignment scores generated by the above procedures. We call this kind of Z-scores composition-corrected Z-scores and let Z_{comp} denote it. To further cancel out the bias caused by the number of alternative sequence-template alignments, Bryant and Altschul [13] and PROSPECT-II [6] also used the following procedures:

- Shuffle the whole sequence residues randomly.
- Find the optimal alignment between the shuffled sequence and the template and calculate the alignment score.
- Repeat the above two steps as many as 100 times.

The final Z-score is the alignment score in standard deviation units relative to the mean alignment score. We call it alignment-number-corrected Z-score and use Z_{raw} to denote it.

The Z-score method has the following two drawbacks:

1. It takes a lot of extra time to calculate Z-scores, especially the alignment-number-corrected Z-scores. In order to calculate the alignment-number-corrected Z-score for each threading pair, the target sequence has to be shuffled and threaded many times. In order to save time, many prediction programs like PROSPECT-I [5] only calculate the composition-corrected Z-score. Even though this, the computational efficiency hinders the Z-score method from genome-scale structure prediction.
2. Z-score is hard to interpret, especially when the scoring function is the weighted sum of various energy items such as mutation score, environmental fitness score, pairwise score, secondary structure score, gap penalty, and score induced from NMR data. For example, when the sequence is shuffled, shall we shuffle the position specific profile information and the predicted secondary structure type at each sequence residue? If we choose to shuffle the secondary structure, then the shuffled secondary structure arrangement does not look like a protein's. Otherwise, if we choose to predict the secondary structure again, the whole process will take a very long time.

In our previous paper [10], we have very briefly introduced a binary SVM classification method for template selection. Although classification-based methods run much faster and have better sensitivity than the Z-score method, they still have some problems. The similarity between two proteins could be at fold level, superfamily level or family level. The binary SVM classification method can only treat the three different similarity levels as a single one, which will lose some important information. Multiclass SVM cannot be directly used here since the relationship among the three similarity levels is hierarchy. If a sequence-template pair is recognized by a classifier similar at a family-level, then this pair is also similar at a superfamily-level and fold-level. It would be hard to build a classifier for those sequence-template pairs which are similar at superfamily-level but not family-level. A single binary classifier cannot effectively differentiate one similarity level from another. Generally speaking, the closer the relationship between two proteins is, the better alignment the two proteins have. That is, statistically, two proteins similar at a family-level will have a better alignment than two proteins similar at a superfamily-level, which in turn better than two proteins similar at a fold-level. By using SVM regression method with the alignment accuracy as the objective function, we can differentiate these three similarity levels in a single SVM model. Another problem is that there are some classification errors in the protein fold classification databases such as SCOP [15] and CATH [16], [17] and the classification criteria are also different, which will introduce systematic errors in training classifiers. Finally, the most important problem is that even if SVM classification can predict two proteins to have a similar fold, it is possible that the alignment accuracy between them is really bad. As mentioned before, what we need is one template with the best alignment to the target sequence. Classification-based methods can only recognize those templates with a similar fold as the target sequence, but cannot tell which template has the best alignment to the target sequence. A template with a similar fold as the target sequence cannot guarantee a good alignment to the sequence unless the structures of both proteins are available. The preferred result is that the better alignment the template has to the target sequence, the better the rank of the template. Therefore, the template selection problem (or the fold recognition problem in this context) is not a classification problem, but a ranking problem.

In [18], Ding and Dubchak have studied how to use multiclass SVM and neural network to do protein fold recognition. However, the problem addressed in this paper is different from that addressed in Ding and Dubchak's paper. Ding and Dubchak's paper describes how to classify one protein based on only its amino acid sequence. In principle, Ding et al.'s method can return all the protein structures from PDB with a similar fold as a given sequence, but not the template with the best alignment to the target sequence. The problem addressed in this paper is how to choose the template with the best alignment to a given target sequence based on all the sequence-template alignments generated by a threading program. A threading program not only makes use of the sequence information of

both the target sequence and the template, but also takes into consideration the structure information of the template. Our final goal is to build the three-dimensional structure for a given protein sequence based on the chosen sequence-template alignment rather than to do fold classification. Another major difference is that Ding et al.'s approach has to train a large number of SVM models in order to scale up to the practical application while our method only needs one or two SVM models.

In this paper, we will introduce an SVM regression approach to directly predict the alignment accuracy of a given sequence-template alignment. The predicted alignment accuracy has a correlation coefficient 0.71 with the real alignment accuracy. Then, we use the predicted sequence-template alignment accuracy to rank all the templates for a given sequence. Experimental results show that the predicted alignment accuracy has a much better sensitivity and specificity than composition-corrected Z-score method and a much better computational efficiency. SVM regression is also better than SVM classification and the alignment number-corrected Z-score method in terms of sensitivity. In addition, the alignment accuracy is also easier to interpret than the classification results.

The rest of this paper is organized as follows: Section 2 will briefly introduce the SVM regression method. In Section 3, we will describe how to generate all the features used by SVM models from each sequence-template alignment. Section 4 describes several experiment results and compares SVM regression with the Z-score method and SVM classification method. Finally, Section 5 draws some conclusions.

2 SVM Regression

In this section, we briefly introduce the linear and nonlinear Support Vector Machine (SVM) regression methods. Support Vector Machines are developed in the late 1970s by Vapnik [19]. The most commonly used SVM is the nonlinear SVM. However, we will start with the linear SVM regression because the nonlinear SVM is just a kernelized linear SVM. Smola and Schölkopf [20] provides an excellent tutorial on SVM regression.

2.1 Linear SVM Regression

Given a set of training data $\{x_i, y_i\}$, $i = 1, 2, \ldots, l$, $y_i \in R$, and $x_i \in R^m$, we call x_i the input data point, and y_i the observed response given an input x_i. Our goal is to find a function f(x) that has at most ϵ deviation from the observed response. Suppose that the relationship between x and y is linear. That is, there is a vector $w \in R^m$ such that $f(x) = wx + b$. There might be multiple w satisfying this equation, so we require that w has the smallest Euclidean norm to guarantee a unique w. Therefore, we can write this problem as the following optimization problem:

$$\min \frac{1}{2}||w||^2$$

subject to

$$\begin{aligned} y_i - wx_i - b &\le \epsilon \\ wx_i + b - y_i &\ge \epsilon. \end{aligned}$$

It is not always possible to guarantee such a $f(x)$ exists. In order to have a feasible solution, we allow for some errors. That is, we introduce slack variable ζ_i and ζ_i^* $(i = 1, 2, \ldots, l)$ to achieve the following optimization problem:

$$\min \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}(\zeta_i + \zeta_i^*)$$

subject to

$$\begin{aligned} y_i - wx_i - b &\le \epsilon + \zeta_i \\ wx_i + b - y_i &\ge \epsilon + \zeta_i^* \\ \zeta_i, \zeta_i^* &\ge 0, \end{aligned}$$

where C is the penalty factor.

By introducing Lagarangian multiplier λ_i and λ_i^* $(i = 1, 2, .., l)$ for the constraints, we have the following dual problem:

$$\max L_D = -\frac{1}{2}\sum_{i,j=1}^{l}(\lambda_i - \lambda_i^*)(\lambda_j - \lambda_j^*)(x_i x_j) - \epsilon\sum_{i=1}^{l}(\lambda_i + \lambda_i^*) + \sum_{i=1}^{l} y_i(\lambda_i - \lambda_i^*)$$

subject to

$$\begin{aligned} \sum_{i=1}^{l}(\lambda_i - \lambda_i^*) &= 0 \\ \lambda_i, \lambda_i^* &\in [0, C]. \end{aligned}$$

After solving λ_i and λ_i^*, we have:

$$f(x) = \sum_{i=1}^{l}(\lambda_i - \lambda_i^*)(x_i x) + b.$$

2.2 Nonlinear SVM Regression

Now, we generalize the linear SVM to accommodate the case where the observed outputs are not a linear function of the input data. A very straightforward idea is to map the data points into a higher dimension space and then do linear regression in the higher dimension space. the only difference lies in that in the objective function L_D, we replace $(x_i x_j)$ with $\phi(x_i)\phi(x_j)$, where ϕ is the mapping function. Theoretically, there is no problem if we know the mapping function ϕ. However, there is a computational challenge if we calculate $\phi(x_i)$ directly, when its dimension is very large, say millions of dimensions or infinite. Notice that in L_D, only the products $\phi(x_i)\phi(x_j)$ but not any $\phi(x_i)$ are needed. In order to circumvent this difficulty, the mapping function ϕ is chosen such that the inner product of any two points in the new space can be represented as a function of the original two points. That is, there is a function K such that $\phi(x_i)\phi(x_j) = K(x_i, x_j)$. Then, we do not need to directly calculate $\phi(x_i)$ and $\phi(x_i)\phi(x_j)$ because we only need to compute $K(x_i, x_j)$. Function K is also called a kernel function.

3 Features

After the optimal sequence-template alignment is generated by a threading algorithm like the dynamic programming

algorithm in GenTHREADER [6] or the linear programming algorithm in RAPTOR [10], we extract some features from it. In calculating the features, we take the evolutionary information of sequences and templates into account. For each template, we use PSI-BLAST to generate the position specific mutation matrix $PSSM$. $PSSM(i,a)$ denotes the mutation score of residue a at template position i. It is defined as the log-odds of residue a occurring at position i. We also use PSI-BLAST to generate the position specific frequency matrix $PSFM$ for each target sequence. $PSFM(j,b)$ denotes the occurring frequency of residue b at sequence position j. Both $PSSM$ and $PSFM$ are used in our feature calculation. Let $A(i)$ denote the aligned sequence position of template position i. If the template position i is not aligned to any sequence position, then $A(i)$ is invalid. In this section, if $A(i)$ is invalid, then $PSFM(A(i),a)$ and $PSSM(A(i),a)$ are 0 for any i and a.

3.1 Mutation Score

Mutation score measures the sequence similarity between the target protein and the template protein. At each template position i, the mutation score is $\sum_a PSFM(A(i),a) \times PSSM(i,a)$. So, the total mutation score can be calculated by the following equation:

$$E_m = \sum_i \sum_a PSFM(A(i),a) \times PSSM(i,a).$$

3.2 Sequence Identity

In addition to mutation score, we also use the number of identical residues in the alignment to measure the sequence similarity from another aspect. Although low sequence identity is not so useful in identifying the relationship between two proteins, but high sequence identity can indicate that two proteins should be in a similarity level.

3.3 Environmental Fitness Score

At each template position i, we use the following two types of local structural features to describe its environment env_i.

1. Secondary structure type. Secondary structure describes the local conformation of a protein segment. There are three types of secondary structure: α-helix, β-strand (β-sheet) and irregular structure (loop).
2. Solvent accessibility (sa). Three levels are defined: buried (inaccessible), intermediate, and accessible. The boundaries between the different solvent accessibility levels are determined by the Equal-Frequency discretization method. The calculated boundaries are at 7 percent and 37 percent.

The combination of these two local structure features yields nine local structural environments at each template position. Let $F(env,a)$ denote the environment fitness potential for a particular combination of amino acid type a and environment descriptor env. $F(env,a)$ is taken from PROSPECT-II [6].

The total fitness score can be calculated as follows:

$$E_s = \sum_i \sum_a PSFM(A(i),a) \times F(env_i,a).$$

3.4 Pairwise Contact Score

Let $E(i_1,i_2)$ indicate if there is one contact between two template positions i_1 and i_2. We say there is one contact between two positions if and only if the distance between the side chain centers of these two positions are no more than $7\dot{A}$. The pairwise score is calculated as follows:

$$E_p = \sum_{i_1<i_2} E(i_1,i_2) \sum_a \{PSFM(A(i_1),a) \times \sum_b PSFM(A(i_2),b)P(a,b)\},$$

where $P(a,b)$ denotes the pairwise contact potential between two residues a and b. P is taken from PROSPECT-I [5].

3.5 Secondary Structure Score

Let $SS(i,A(i))$ denote the secondary structure difference between the template position i and the sequence position $A(i)$. We use PSIPRED [21] or other secondary structure predictors to predict the secondary structure of the query sequence. Let $\alpha(j)$, $\beta(j)$, and $loop(j)$ denote the predicted confidence levels of α-helix, β-sheet and loop at sequence position j, respectively. If the secondary structure type at template position i is α-helix, then $SS(i,A(i)) = \alpha(A(i)) - loop(A(i))$. Otherwise, if the secondary structure type at template position i is β-sheet, then $SS(i,A(i)) = \beta(A(i)) - loop(A(i))$. The total secondary structure score is calculated as follows:

$$E_{ss} = \sum_i SS(i,A(i)).$$

3.6 Gap Penalty

In order to guarantee the quality of sequence-template alignments, some gaps must be allowed in the alignment. However, if there are too many gaps, especially gap openings, in the sequence-structure alignment, then it might indicate that the quality of this alignment is bad. The gap penalty function is assumed to be an affine function $b + ge$, that is, a gap open penalty b plus a length-dependent gap extension penalty ge where g is the gap length. In our scoring function, b is set at 10.6 and e at 0.8 per single gap.

3.7 Contact Capacity Score

Contact capacity potential accounts for the hydrophobic contribution of free energy. Contact capacity characterizes the capability of a residue making a certain number of contacts with any other residues in a single protein. Let $CC(a,k)$ denote the contact potential of amino acid type a having k contacts. $CC(a,k)$ can be calculated by the following equation:

$$CC(a,k) = -\log \frac{N(a,k)}{N(k)N(a)/N},$$

where $N(a,k)$ is the number of residues of type a and with k contacts, $N(k)$ the number of residues having k contacts, $N(a)$ the number of residues of type a, and N the total number of residues. The total contact capacity score can be calculated as follows:

$$E_c = \sum_i \sum_a PSFM(A(i), a) \times CC(a, CN(i)),$$

where $CN(i)$ denotes the number of contacts at template position i.

3.8 Alignment Topology

Besides the above-mentioned alignment scores, we also extract the following features to describe the alignment topology:

1. Alignment length: the number of aligned residues. If two large proteins have only very short alignment length, then it very unlikely that these two proteins have similar structures.
2. The number of aligned contacts: the number of template contacts with two ends being the aligned residues. The larger the protein, the more contacts it has. So, this feature, together with the alignment length, can indicate if the aligned sequence can form an independent protein domain.
3. The number of unaligned contacts: the number of template contacts with one end being unaligned. If this number is big relative to the alignment length, then it may indicate that the aligned part is not an independent domain.

3.9 Z-Score

As mentioned in Section 1, Z-score is hard to interpret when the scoring function contains weight factors. Calculation of Z_{raw} is also time-consuming. Here, we only calculate the composition-corrected Z-scores. In addition to Z_{comp}, we also calculate the composition-corrected Z-scores of the following individual score items: Z-score of the mutation score Z_m, Z-score of the fitness score Z_s, Z-score of the pairwise score Z_p, Z-score of secondary structure score Z_{ss}, and Z-score of the contact capacity score Z_c. They are defined by the following equations:

$$Z_m = \frac{\bar{E}_m - E_m}{\sigma(E_m)},$$
$$Z_s = \frac{\bar{E}_s - E_s}{\sigma(E_s)},$$
$$Z_p = \frac{\bar{E}_p - E_p}{\sigma(E_p)},$$
$$Z_{ss} = \frac{\bar{E}_{ss} - E_{ss}}{\sigma(E_{ss})},$$
$$Z_c = \frac{\bar{E}_c - E_c}{\sigma(E_c)},$$

where $\bar{x}$ is the mean of x and $\sigma(x)$ the standard deviation of x. All score distributions are generated by randomly shuffling the aligned sequence residues.

3.10 Alignment Accuracy

Alignment accuracy is not one of the features extracted from the sequence-structure alignment, but it serves as the objective function of SVM regression. We use SARF [22] to generate the correct alignment between the target protein and the template protein. The alignment accuracy of threading is defined to be the number of correctly aligned positions, based on the correct alignment generated by SARF. A position is correctly aligned only if its alignment position is no more than four position shifts away from its correct alignment.

4 SVM Approach to Fold Recognition

In order to train the SVM model, we randomly choose 300 templates from our template database and 200 sequences from the Holm and Sander's test set [23]. A set of 60,000 training data is formed by threading each of 200 sequences to each of 300 templates. For the purpose of evaluation, all the proteins used in training and test have known structures in PDB and known class labels in SCOP [15]. The alignment accuracy between two proteins with known structures is calculated by SARF. For each feature in this training set, we calculate its mean and standard deviation, and then use them to normalize all the training and test data sets involved in our experiments. We also use Daniel Fischer et al.'s benchmark [24] to fix the parameters and the kernel function of the SVM models. This benchmark contains approximately 70 sequences and 300 templates. Experimental results show that the RBF kernel is best for our SVM models. Finally, we use Lindahl and Elofsson's benchmark [25] as the test set to measure the generalization performance of the SVM models. The Lindahl et al.'s benchmark contains 976 sequences and 976 templates, which lead to 976×975 threading pairs.

Given one test threading pair, our SVM regression model outputs a real value as the predicted alignment accuracy. The output is used to rank all the templates for one target sequence. Given a threading pair, we also calculate its confidence score, which is defined as the predicted alignment accuracy in standard deviation units relative to the mean predicted alignment accuracy of all the threading pairs with the same sequence.

In this section, we compare the performance of Z-score, SVM classification and SVM regression in terms of fold classification, alignment accuracy and specificity. The fold classification accuracy can be easily evaluated by referring to the SCOP database [15]. Several papers [25], [6], [8] have used fold classification accuracy to evaluate the performance of different protein structure prediction programs. Recent CASPs [1], [2], [3] use alignment accuracy to evaluate the performance of different prediction groups since alignment accuracy is more important and sometimes it is hard to differentiate two groups by using only fold classification accuracy. In drawing the sensitivity-specificity curves, we use the fold classification accuracy rather than alignment accuracy as the measure for sensitivity since it is hard to decide the alignment accuracy cutoff based on which we can tell if a sequence-template pair has a similar fold or not. In addition, since the hardness of recognizing family-level similarity, superfamily-level similarity and fold-level similarity is very different, we divide all the sequence-template pairs into four different groups in terms of their relationship: family-level, superfamily-level, fold-level, and others.

4.1 Experiment I

In this experiment, we use the following features to train and test the SVM models:

TABLE 1
The Sensitivity of RAPTOR on Lindahl et al.'s Benchmark

	Family		Superfamily		Fold	
method	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5
SVM regression	83.0	87.6	51.7	65.6	31.6	56.4
SVM classification	84.4	88.3	52.2	65.6	32.5	56.1
Z_{comp}	74.0	80.2	36.8	55.3	17.7	38.2

Z_{Comp} is the composition-corrected Z score.

1. sequence size,
2. template size,
3. alignment length,
4. sequence identity,
5. the number of aligned contacts,
6. the number of unaligned contacts,
7. alignment score,
8. mutation score,
9. environment fitness score,
10. gap penalty,
11. secondary structure score,
12. pairwise contact score, and
13. contact capacity score.

In training the SVM regression model, the alignment accuracy is used as the objective function. We fix the parameters of our SVM regression model such that the sensitivity of our model on the Fischer et al.'s benchmark is the highest. For this benchmark, the correlation coefficient between the predicted alignment accuracy and the real alignment accuracy is 0.72. We randomly choose 15,000 threading pairs from the Lindahl et al.'s benchmark and use SARF to calculate their real alignment accuracy. For these 15,000 threading pairs, the correlation coefficient between the predicted alignment accuracy and the real alignment accuracy is 0.71. This indicates that the generalization performance of our SVM regression model is very good.

In order to compare the performance of SVM regression with that of SVM classification. We also train a SVM classification model by using the same data. A threading pair is considered a positive example only if the sequence and the template in the same fold class according to SCOP database [15]. SVM classification was used in RAPTOR [10] and achieved a very good performance in CAFASP3 [26]. Table 1 shows the sensitivity of SVM regression method on the Lindahl et al.'s benchmark at all similarity levels. In calculating the top 1 sensitivity, only the first-ranked sequence-template alignment is considered. In calculating the top 5 sensitivity, the best sequence-template alignment among the first five is considered. The similarity relationship between two proteins is judged based on the SCOP database. As shown in Table 1, the sensitivity of SVM regression method is much better than that of the composition-corrected Z score and similar to that of SVM classification method.

TABLE 2
The Sensitivity of RAPTOR on Lindahl et al.'s Benchmark at Three Different Similarity Levels

	Family		Superfamily		Fold	
method	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5
SVM regression	86.6	89.3	56.3	69.0	38.2	58.7
SVM classification	83.1	87.9	51.9	67.1	32.5	50.4
Z_{comp}	74.0	80.2	36.8	55.3	17.7	38.2
PROSPECT-II	84.1	88.2	52.6	64.8	27.7	50.3

All composition-corrected Z-scores are used as features.

4.2 Experiment II

In addition to the features used in Experiment I, we also incorporate all the composition-corrected Z-scores such as Z_{comp}, Z_m, Z_s, Z_p, Z_{ss}, and Z_c into our feature space to see if the SVM regression can improve its sensitivity further. For the Fischer's et al. benchmark, the correlation coefficient between the predicted alignment accuracy and the real alignment accuracy is also 0.72. By using the same set of randomly chosen 15,000 threading pairs from the Lindahl et al.'s benchmark, we achieve the same correlation coefficient between the predicted alignment accuracy and the real alignment accuracy. The result in Table 2 demonstrates that composition-corrected Z-scores are helpful for the SVM regression method, but not SVM classification. The sensitivity of SVM regression is better than that of SVM classification at all the similarity levels, especially at the fold level. But, it takes a lot of time to calculate all the composition-corrected Z scores. Please notice that there is no big difference in terms of computational time between calculating a single Z_{comp} and calculating Z_{comp}, Z_m, Z_s, Z_p, Z_{ss}, and Z_c all together. In this table, we also list the sensitivity of PROSPECT-II [6], which uses the alignment-number-corrected Z-score to rank all the templates. As shown in this table, SVM regression performs better than PROSPECT-II at all the similarity levels and much better at the fold level.

4.3 Experiment III

In order to achieve as high sensitivity as that in Experiment II without taking a lot of time to calculate the composition-corrected Z-scores of all the threading pairs, we first use the SVM regression model in Experiment I to rank all the templates, then choose the top N $(N = 20, 30, \ldots, 100)$ templates for each sequence and, finally, use the SVM regression model in Experiment II to rerank the chosen N templates, which means we only need to calculate the composition-corrected Z-scores for the top N threading pairs. As shown in Table 3, only a very small fraction (50 out of 975) of threading pairs need Z-scores to achieve the same sensitivity as in Experiment II. Notice that SVM prediction can be done very quickly after the SVM model is trained. Therefore, we can achieve the best sensitivity with only little extra efforts.

TABLE 3
The Sensitivity of RAPTOR on Lindahl et al.'s Benchmark

N	Fold Level	Superfamily Level	Family Level
20	0.359	0.555	0.864
30	0.365	0.555	0.864
40	0.373	0.558	0.864
50	0.379	0.562	0.866
60	0.379	0.562	0.866
70	0.379	0.562	0.866
80	0.379	0.562	0.866
90	0.379	0.562	0.866
100	0.379	0.562	0.866
976	0.382	0.562	0.866

The top N templates are chosen by the SVM regression model in Experiment I and reranked by the SVM regression model in Experiment II. Only top 1 sensitivity is shown in this table.

4.4 Alignment Accuracy

As mentioned before in this paper, our ultimate goal is to rank all the sequence-template alignments for a given target sequence such that the first-ranked sequence-template alignment has the best alignment accuracy. In order to compare SVM classification and SVM regression approaches in terms of alignment accuracy, for a given target sequence, we rank all the candidate sequence-template alignments by our trained SVM classification model and SVM regression model, respectively. Then, we calculate the following three types of average alignment accuracy.

1. For each sequence, we obtain the first-ranked sequence-template alignment and then calculate the average alignment accuracy over all the sequences. We call this average alignment accuracy family-level alignment accuracy.
2. We first remove all the sequence-template pairs similar at a family level. Then, we obtain the first-ranked sequence-template alignment for each sequence and calculate the average alignment accuracy over all the sequences. We call this average alignment accuracy superfamily-level alignment accuracy.
3. We remove all the sequence-template pairs similar at a superfamily level or family level. Then we obtain the first-ranked sequence-template alignment for each sequence and calculate the average alignment accuracy over all the sequences. We call this average alignment accuracy fold-level alignment accuracy.

The average alignment accuracy is listed in Table 4. This table clearly indicates that the average alignment accuracy can be improved by 30 percent at fold level, 25 percent at superfamily level, and 10 percent at family level if we replace SVM classification with SVM regression. Please note that for a portion of target sequences, both SVM classification and regression methods fail to rank a reasonable sequence-template alignment at top. Therefore, the average alignment accuracy shown in this table is fairly low.

TABLE 4
Comparison of SVM Classification and SVM Regression Methods in Terms of Alignment Accuracy

method	Fold Level	Superfamily Level	Family Level
SVM classification	13.8	20.6	49.8
SVM regression	17.3	25.5	55.9

To further compare SVM regression and SVM classification approaches, we randomly sample 490 sequences from the Lindahl's benchmark and calculate the average alignment accuracy using the method described in this subsection. Fig. 1 illustrates the alignment accuracy improvement ratio by our SVM regression model over our SVM classification model for 100 samplings. The improvement ratio is defined as the alignment accuracy gain by SVM regression over SVM classification divided by the alignment accuracy of SVM classification. As shown in this figure, SVM regression approach is always better than SVM classification in terms of alignment accuracy. The mean (standard deviation) of the improvement ratio is $0.2517(0.0441)$ at fold level, $0.2364(0.0388)$ at superfamily level, and $0.1206(0.0202)$ at family level, respectively.

4.5 Specificity

We further examine the specificity of the SVM regression model in Experiment II. All threading pairs are ranked by the confidence score and the sensitivity-specificity curves are drawn in Figs. 2, 3, and 4. The sensitivity-specificity curve describes the quality of the confidence score. Figs. 2 and 3 demonstrate that SVM regression method is much better than the composition-corrected Z-score method and a little better than SVM classification. At the family level, SVM regression achieves a sensitivity of 45.6 percent and 73.6 percent at

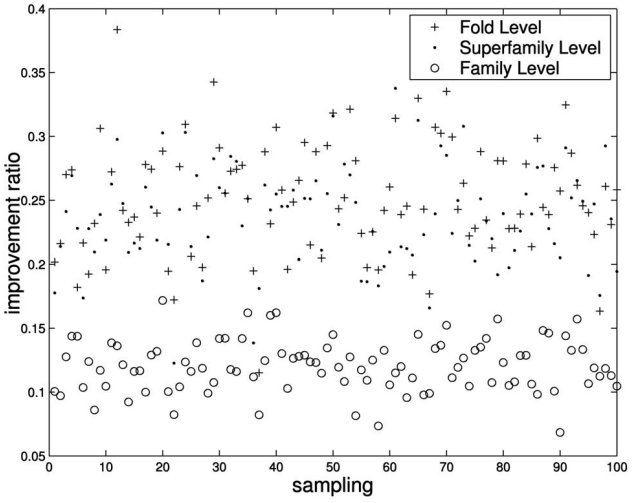

Fig. 1. Performance comparison between SVM regression and SVM classification in terms of alignment accuracy.

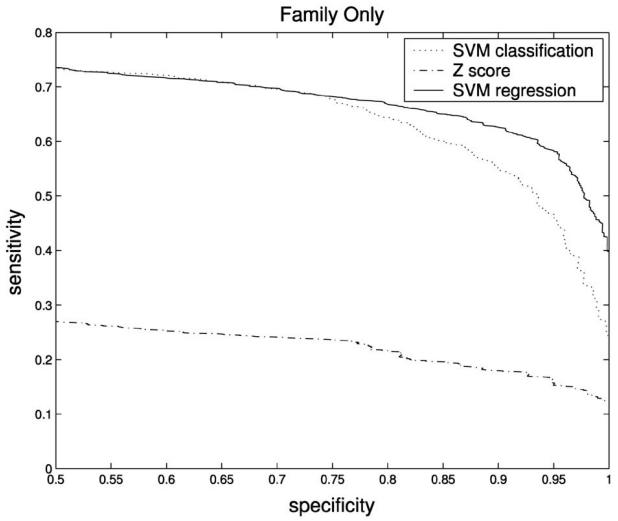

Fig. 2. Family-level specificity-sensitivity curves on the Lindahl's benchmark set.

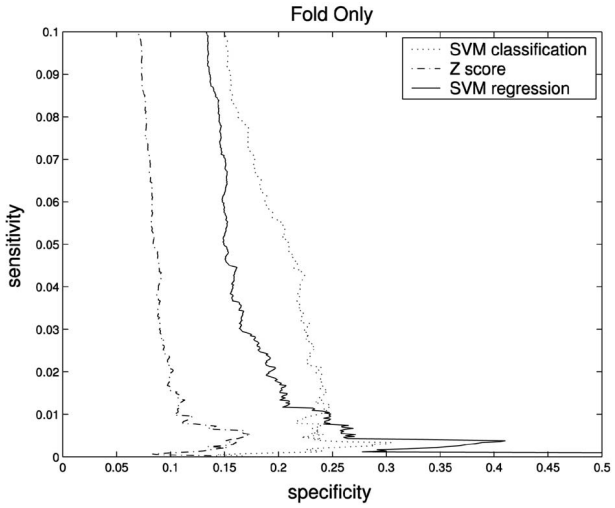

Fig. 4. Fold-level specificity-sensitivity curves on the Lindahl's benchmark set.

99 percent and 50 percent specificities, respectively, whereas SVM classification achieves a sensitivity of 29.0 percent and 73.5 percent at 99 percent and 50 percent specificities, respectively. At the superfamily level, SVM regression has a sensitivity of 4.5 percent and 19.6 percent at 99 percent and 50 percent specificities, respectively. In contrast, SVM classification has a sensitivity of 2.4 percent and 16.5 percent at 99 percent and 50 percent specificities, respectively. Fig. 4 shows that at the fold level, SVM regression is still much better than the composition-corrected Z-score and there is no big difference between SVM regression method and SVM classification method.

5 Conclusions

In this paper, we have proposed a SVM regression method to predict the alignment accuracy of a threading pair, which is used to do fold recognition. Experimental results show that SVM regression method has much better performance in both sensitivity and specificity than the composition-corrected Z score method. SVM regression method also performs better than SVM classification method. In addition, SVM regression method enables the threading program to run faster since only a very small portion of threading pairs need to calculate the composition-corrected Z-scores. The drawback of SVM regression method is that if we change the formula in calculating any feature, then we have to retrain our SVM models, which is the drawback of any machine learning based method. However, the training time is affordable since we only need to train our SVM models once whenever the feature space is changed. The future work is to extend the SVM regression method to predict other quality indices such as MaxSub score [27], which is used as the evaluation criteria of CAFASP3 [26].

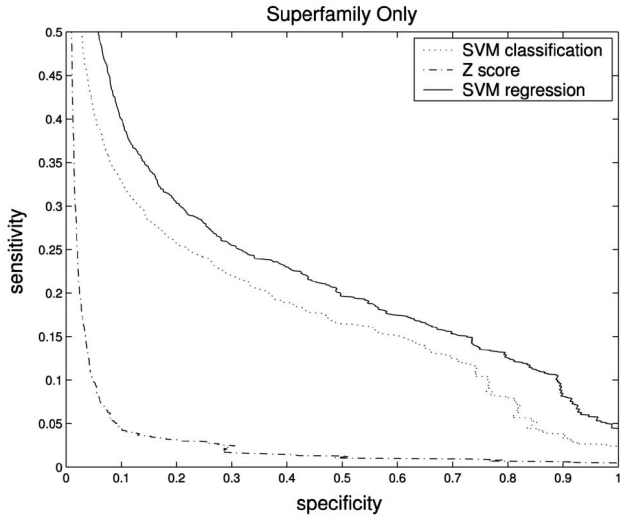

Fig. 3. Superfamily-level specificity-sensitivity curves on the Lindahl's benchmark set.

References

[1] J. Moult, T. Hubbard, F. Fidelis, and J. Pedersen, "Critical Assessment of Methods on Protein Structure Prediction (CASP)-Round III," *Proteins: Structure, Function and Genetics,* vol. 37, pp. 2-6, Dec. 1999.

[2] J. Moult, F. Fidelis, A. Zemla, and T. Hubbard, "Critical Assessment of Methods on Protein Structure Prediction (CASP)-Round IV," *Proteins: Structure, Function and Genetics,* vol. 45, pp. 2-7, Dec. 2001.

[3] J. Moult, F. Fidelis, A. Zemla, and T. Hubbard, "Critical Assessment of Methods on Protein Structure Prediction (CASP)-Round V," *Proteins: Structure, Function and Genetics,* vol. 53, pp. 334-339, Oct. 2003.

[4] A. Sali and T.L. Blundell, "Comparative Protein Modelling by Satisfaction of Spatial Restraints," *J. Molecular Biology,* vol. 234, pp. 779-815, 1993.

[5] Y. Xu, D. Xu, and E.C. Uberbacher, "An Efficient Computational Method for Globally Optimal Threadings," *J. Computational Biology,* vol. 5, no. 3, pp. 597-614, 1998.

[6] D. Kim, D. Xu, J. Guo, K. Ellrott, and Y. Xu, "PROSPECT II: Protein Structure Prediction Method for Genome-Scale Applications," *Protein Engineering,* vol. 16, no. 9, pp. 641-650, 2003.

[7] L.A. Kelley, R.M. MacCallum, and M.J. E. Sternberg, "Enhanced Genome Annotation Using Structural Profiles in the Program 3D-PSSM," *J. Molecular Biology,* vol. 299, pp. 499-520, 2000.

[8] J. Shi, L.B. Tom, and M. Kenji, "FUGUE: Sequence-Structure Homology Recognition Using Environment-Specific Substitution Tables and Structure-Dependent Gap Penalties," *J. Molecular Biology,* vol. 310, pp. 243-257, 2001.

[9] D.T. Jones, "GenTHREADER: An Efficient and Reliable Protein Fold Recognition Method for Genomic Sequences," *J. Molecular Biology,* vol. 287, pp. 797-815, 1999.
[10] J. Xu, M. Li, D. Kim, and Y. Xu, "RAPTOR: Optimal Protein Threading by Linear Programming," *J. Bioinformatics and Computational Biology,* vol. 1, no. 1, pp. 95-117, 2003.
[11] T. Akutsu and S. Miyano, "On the Approximation of Protein Threading," *Theoretical Computer Science,* vol. 210, pp. 261-275, 1999.
[12] D.T. Jones, W.R. Taylor, and J.M. Thornton, "A New Approach to Protein Fold Recognition," *Nature,* vol. 358, pp. 86-98, 1992.
[13] S.H. Bryant and S.F. Altschul, "Statistics of Sequence-Structure Threading," *Current Opinions in Structural Biology,* vol. 5, pp. 236-244, 1995.
[14] Y. Xu, D. Xu, and V. Olman, "A Practical Method for Interpretation of Threading Scores: An Application of Neural Networks," *Statistica Sinica,* special issue on bioinformatics, vol. 12, pp. 159-177, 2002.
[15] A.G. Murzin, S.E. Brenner, T. Hubbard, and C. Chothia, "SCOP: A Structural Classification of Proteins Database for the Investigation of Sequences and Structures," *J. Molecular Biology,* vol. 247, pp. 536-540, 1995.
[16] C.A. Orengo, A.D. Michie, S. Jones, D.T. Jones, M.B. Swindells, and J.M. Thornton, "CATH-A Hierarchic Classification of Protein Domain Structures," *Structure,* vol. 5, pp. 1093-1108, 1997.
[17] F.M. G. Pearl, D. Lee, J.E. Bray, I. Sillitoe, A.E. Todd, A.P. Harrison, J.M. Thornton, and C.A. Orengo, "Assigning Genomic Sequences to CATH," *Nucleic Acids Research,* vol. 28, pp. 277-282, 2000.
[18] C.H.Q. Ding and I. Dubchak, "Multi-Class Protein Fold Recognition Using Support Vector Machine and Neural Networks," *Bioinformatics,* vol. 17, no. 4, pp. 349-358, 2001.
[19] V.N. Vapnik, *The Nature of Statistical Learning Theory.* Springer, 1995.
[20] A.J. Smola and B. Schölkopf, "A Tutorial on Support Vector Regression," technical report, Oct. 1998.
[21] D.T. Jones, "Protein Secondary Structure Prediction Based on Position-Specific Scoring Matrices," *J. Molecular Biology,* vol. 292, pp. 195-202, 1999.
[22] N.N. Alexandrov, "SARFing the PDB," *Protein Eng.,* vol. 9, pp. 727-732, 1996.
[23] L. Holm and C. Sander, "Decision Support System for the Evolutionary Classification of Protein Structures," *Proc. Conf. Intelligent Systems for Molecular Biology (ISMB),* vol. 5, pp. 140-146, 1997.
[24] D. Fischer, A. Elofsson, J.U. Bowie, and D. Eisenberg, "Assessing the Performance of Fold Recognition Methods by Means of a Comprehensive Benchmark," *Biocomputing: Proc. 1996 Pacific Symp.,* pp. 300-318, 1996.
[25] E. Lindahl and A. Elofsson, "Identification of Related Proteins on Family, Superfamily and Fold Level," *J. Molecular Biology,* vol. 295, pp. 613-625, 2000.
[26] D. Fischer, L. Rychlewski, R.L. Dunbrack, A.R. Ortiz, and A. Elofsson, "CAFASP3: The Third Critical Assessment of Fully Automated Structure Prediction Methods," *Proteins: Structure, Function and Genetics,* vol. S6, no. 53, pp. 503-516, Oct. 2003.
[27] N. Siew, A. Elofsson, L. Rychlewski, and D. Fischer, "Maxsub: An Automated Measure for the Assessment of Protein Structure Prediction Quality," *Bioinformatics,* vol. 16, no. 9, pp. 776-785, 2000.

Jinbo Xu received the PhD degree from the University of Waterloo in 2003, the MSc degree from the Institute of Computing Technology, Chinese Academy of Sciences in 1999, and the BS degree from the University of Science and Technology of China in 1996. He is currently a postdoctoral fellow with the Department of Mathematics and Computer Science and the Artificial Intelligence Laboratory, Massachusetts Institute of Technology. His current research interests include development of computational methods/software for solving a wide range of biology problems such as protein structure alignment, protein structure prediction, and homology search.