Fold Recognition by Predicted Alignment Accuracy

Jinbo Xu

Jinbo Xu is with School of Computer Science, University of Waterloo, Canada.

Email: j3xu@uwaterloo.ca

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 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-scores is time-consuming and not suitable for genome-scale structure prediction. Z-scores are also hard to interpret when the scoring function is the weighted sum of several energy items of different meanings. This paper presents a Support Vector Machine (SVM) regression approach to directly predict the alignment accuracy of protein threading, 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.

I. INTRODUCTION

Protein structure prediction by protein threading technique has demonstrated a great success in recent CASPs (Critical Assessment of Structure Prediction) [1], [2], [3]. Protein threading makes a structure prediction by finding the optimal alignment between the target sequence and each of the available protein structures (also called templates) in Protein Data Bank (PDB), and then choosing the best overall template as the basis on which the structure of the target sequence is built. The algorithm for finding the optimal sequence-template alignment has been researched extensively [4], [5], [6], [7], [8], [9], [10], [11]. However, how to choose the best template based on alignments (*i.e.*, fold recognition) is also critical to the success of protein threading. Fold recognition requires a criterion to identify the best template for one target sequence. 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 [12]. So far, there are two strategies used by the structure prediction community for fold recognition: recognition based on Z-scores [12], and recognition by machine learning methods [8], [9]. Most of the current prediction programs use the traditional Z-score to recognize the best-fit templates, whereas several programs such as GenTHREADER [8] and PROSPECT-I [13] [1] use a neural network to rank the templates. The neural network method treats the template selection problem as a classification problem. 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 et al. [12] 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 et al. [12] and PROSPECT-II [5] also used the following procedures:

- Shuffle the whole sequence residues randomly;
- Find the optimal alignment between the shuffled sequence and the template and calculate

[1]The draft version of PROSPECT-II [5] also proposed a SVM classification method, but the final version did not include it.

the alignment score;

- Repeat the above two steps many 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: (i) 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 [4] only calculate the composition-corrected Z-score. Even though this, the computational efficiency hinders the Z-score method from genome-scale structure prediction. (ii) 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 [9], we have very briefly introduced the SVM classification method for fold recognition. 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. Classification-based methods can only treat the three different similarity levels as a single one. Multi-class SVM cannot be used here since the relationship among the three similarity levels is hierarchy. That is, if two proteins

are in a family, then they are also in a superfamily and have the same fold. Classification-based methods cannot effectively differentiate one similarity level from another. The other problem is that even if SVM classification can predict two proteins to be similar in at least fold level, it is possible that the alignment accuracy between them is really bad. A template with only the same fold as but not in the same family as the target might be ranked higher than a template in the same family as the target, which is not what we expect. The preferred ranking is that the closer the template to the target, the better the rank of the template.

In this paper, we will introduce a 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 II will briefly introduce the SVM regression method. In Section III, we will describe how to generate all the features used by SVM models from each sequence-template alignment. Section IV describes several experiment results and compares SVM regression with the Z-score method and SVM classification method. Finally, Section V draws some conclusions.

II. 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 1970's by Vapnik [14]. 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 et al. [15] provides an excellent tutorial on SVM regression.

1) Linear SVM regression: Given a set of training data $\{x_i, y_i\}$, $i = 1, 2, ..., l$, $y_i \in R$, 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

$$y_i - wx_i - b \leq \epsilon$$

$$wx_i + b - y_i \geq \epsilon$$

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^* ($i = 1, 2, ..., l$) to achieve the following optimization problem:

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

subject to

$$y_i - wx_i - b \leq \epsilon + \zeta_i$$

$$wx_i + b - y_i \geq \epsilon + \zeta_i^*$$

$$\zeta_i, \zeta_i^* \geq 0$$

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

$$\sum_{i=1}^{l}(\lambda_i - \lambda_i^*) = 0$$

$$\lambda_i, \lambda_i^* \in [0, C]$$

After solving λ_i and λ_i^*, we have:

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

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.

III. FEATURES

After the optimal sequence-template alignment is found by a threading algorithm like the dynamic programming algorithm in GenTHREADER [5] or the linear programming algorithm in RAPTOR [9], we calculate 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) 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:

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

4) 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.

5) 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 ($beta$-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% and 37%.

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 [5].

The total fitness score can be calculated as follows:

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

6) Pairwise Contact Score: Let $E(i_1, i_2)$ indicate if there is one contact between two template position 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 [4].

7) 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 [16] 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:

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

8) 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.

9) 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.

10) 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.

11) Z-score: As mentioned in Section I, 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:

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

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.

12) 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 [17] 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.

IV. 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 [18]. A set of 60,000 training data is formed by threading each of 200 sequences to each of 300 templates. The alignment accuracy between the sequence and the template is calculated by SARF. We also use Daniel Fischer et al.'s benchmark [19] 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 [20] 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.

A. Experiment I

In this experiment, we use the following features to train and test the SVM models: (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; (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 [21]. SVM classification was used in RAPTOR [9] and achieved a very good performance in CAFASP3 [22]. Table I 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 I, 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 I

THE SENSITIVITY OF RAPTOR ON THE LINDAHL ET AL.'S BENCHMARK. Z_{comp} IS THE COMPOSITION-CORRECTED Z SCORE.

	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

B. 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 1,5000 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 II 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 [5], 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.

TABLE II

THE SENSITIVITY OF RAPTOR ON THE LINDAHL ET AL.'S BENCHMARK AT THREE DIFFERENT SIMILARITY LEVELS. ALL COMPOSITION-CORRECTED Z-SCORES ARE USED AS FEATURES.

	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

C. 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 ($N = 20, 30, ..., 100$) templates for each sequence and finally use the SVM regression model in Experiment II to re-rank 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 IV-C, 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 III

THE SENSITIVITY OF RAPTOR ON THE LINDAHL ET AL.'S BENCHMARK. THE TOP N TEMPLATES ARE CHOSEN BY THE SVM REGRESSION MODEL IN EXPERIMENT I AND RE-RANKED BY THE SVM REGRESSION MODEL IN EXPERIMENT II. ONLY TOP 1 SENSITIVITY IS SHOWN IN THIS TABLE.

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

D. 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 Figure 1, 2 and 3. The sensitivity-specificity curve describes the quality of the confidence score. Figures 1 and 2 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% and 73.6% at 99% and 50% specificities, respectively, whereas SVM classification achieves a sensitivity of 29.0% and 73.5% at 99% and

50% specificities respectively. At the superfamily level, SVM regression has a sensitivity of 4.5% and 19.6% at 99% and 50% specificities, respectively. In contrast, SVM classification has a sensitivity of 2.4% and 16.5% at 99% and 50% specificities respectively. Figure 3 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.

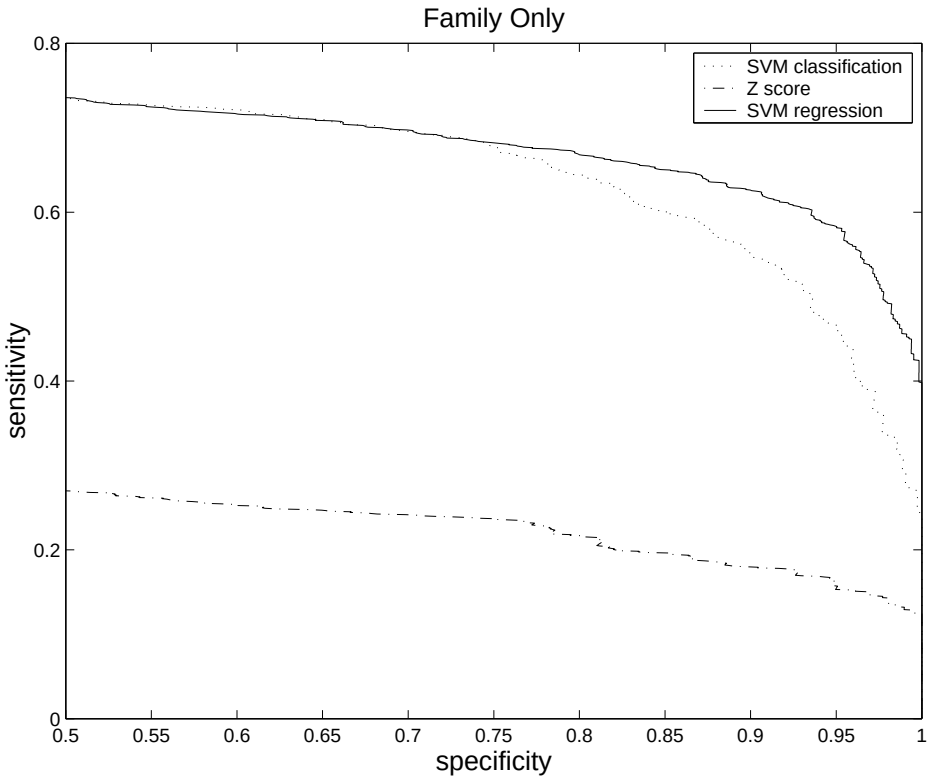

Fig. 1. Family-level specificity-sensitivity curves on the Lindahl's benchmark set.

V. 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

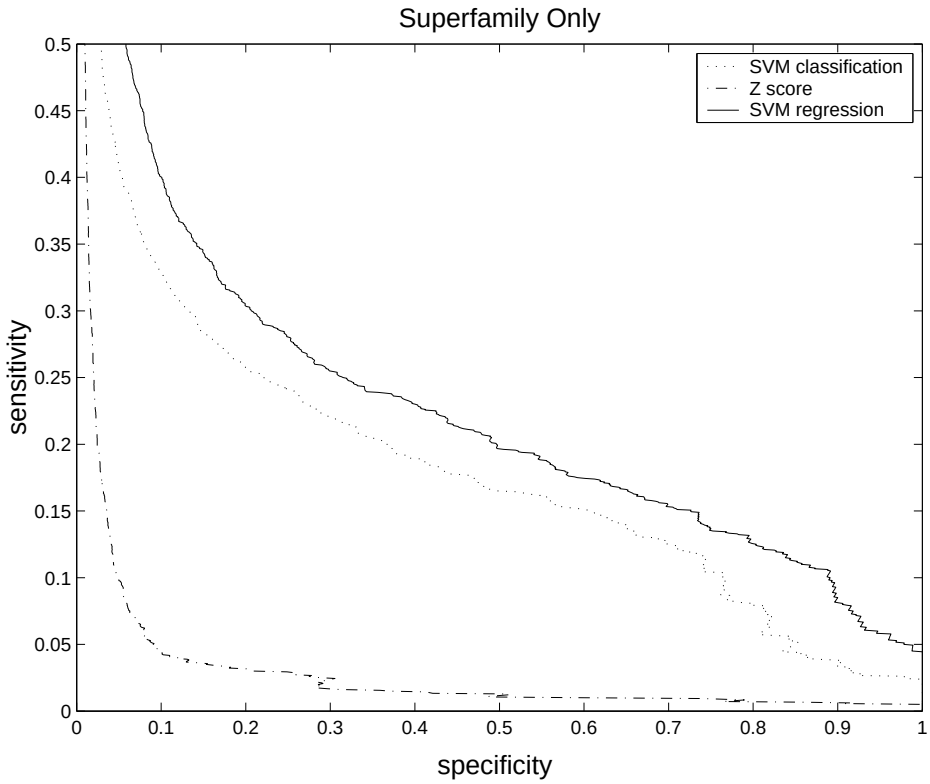

Fig. 2. Superfamily-level specificity-sensitivity curves on the Lindahl's benchmark set.

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 [23],

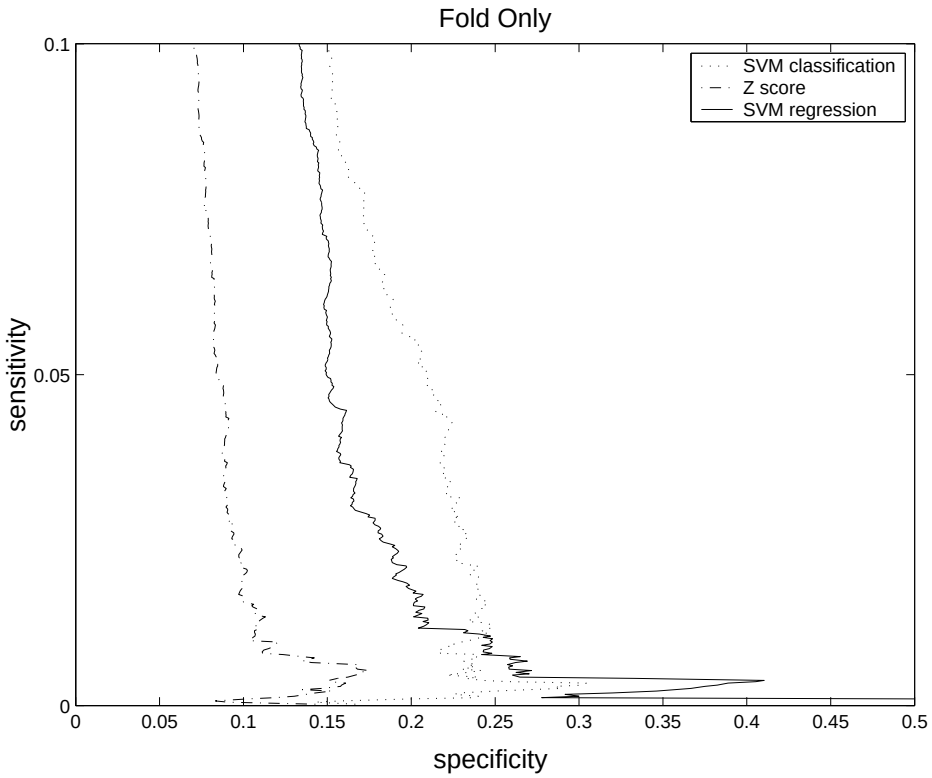

Fig. 3. Fold-level specificity-sensitivity curves on the Lindahl's benchmark set.

which is used as the evaluation criteria of CAFASP3 [22].

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*, 37:2–6, December 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*, 45:2–7, December 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*, 53:334–339, October 2003.

[4] Y. Xu, D. Xu, and E.C. Uberbacher. An efficient computational method for globally optimal threadings. *Journal of Computational Biology*, 5(3):597–614, 1998.

[5] D. Kim, D. Xu, J. Guo, K. Ellrott, and Y. Xu. PROSPECT II: Protein structure prediction method for genome-scale applications. *Protein Engineering*, 16(9):641–650, 2003.

[6] L.A. Kelley, R.M. MacCallum, and M.J.E. Sternberg. Enhanced genome annotation using structural profiles in the program 3D-PSSM. *Journal of Molecular Biology*.

[7] J. Shi, L. B. Tom, and M. Kenji. FUGUE: Sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *Journal of Molecular Biology*.

[8] D.T. Jones. GenTHREADER: An efficient and reliable protein fold recognition method for genomic sequences. *Journal of Molecular Biology*.

[9] J. Xu, M. Li, D. Kim, and Y. Xu. RAPTOR: optimal protein threading by linear programming. *Journal of Bioinformatics and Computational Biology*, 1(1):95–117, 2003.

[10] T. Akutsu and S. Miyano. On the approximation of protein threading. *Theoretical Computer Science*, 210:261–275, 1999.

[11] D.T. Jones, W.R. Taylor, and J.M. Thornton. A new approach to protein fold recognition. *Nature*, 358:86–98, 1992.

[12] S.H. Bryant and S.F. Altschul. Statistics of sequence-structure threading. *Current Opinions in Structural Biology*, 5:236–244, 1995.

[13] 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*, 12:159–177, 2002.

[14] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[15] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. Technical report, October 1998.

[16] D.T. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*.

[17] N.N. Alexandrov. SARFing the PDB. *Protein Engineering*, 9:727–732, 1996.

[18] L. Holm and C. Sander. Decision support system for the evolutionary classification of protein structures. *ISMB*, 5:140–146, 1997.

[19] D. Fischer, A. Elofsson, J.U. Bowie, and D. Eisenberg. Assessing the performance of fold recognition methods by means of a comprehensive benchmark. pages 300–318, Singapore, 1996. Biocomputing: Proceedings of the 1996 Pacific Symposium, World Scientific Publishing Co.

[20] E. Lindahl and A. Elofsson. Identification of related proteins on family, superfamily and fold level. *Journal of Molecular Biology*.

[21] 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. *Journal of Molecular Biology*, 247:536–540, 1995.

[22] 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*, S6(53):503–516, October 2003.

[23] N. Siew, A. Elofsson, L. Rychlewski, and D. Fischer. Maxsub: An automated measure for the assessment of protein structure prediction quality. *Bioinformatics*, 16(9):776–785, 2000.