Finding Conserved Regions in Protein Structures Using Support Vector Machines and Structure Alignment

Tatsuya Akutsu, Morihiro Hayashida, and Takeyuki Tamura

Bioinformatics Center, Institute for Chemical Research, Kyoto University,
Gokasho, Uji, Kyoto 611-0011, Japan
{takutsu,morihiro,tamura}@kuicr.kyoto-u.ac.jp

Abstract. This paper proposes a novel method for finding conserved regions in three-dimensional protein structures. The method combines support vector machines (SVMs), feature selection and protein structure alignment. For that purpose, a new feature vector is developed based on structure alignment for fragments of protein backbone structures. The results of preliminary computational experiments suggest that the proposed method is useful to find common structural fragments in similar proteins.

1 Introduction

Analysis of protein structures is an important topic in bioinformatics and computational biology. In particular, classification of protein structures and identification of common structural patterns are very important. For that purpose, a lot of studies have been done and several databases have been developed such as SCOP [3] and CATH [13]. *Protein structure alignment* is a powerful approach to comparison of protein structures [1,9,16]. Furthermore, *multiple structure alignment* is useful to identify common patterns of multiple protein structures [12,17]. However, it is known that multiple structure alignment and identification of conserved regions are NP-hard if gaps (i.e., insertions and/or deletions of amino acid residues) are allowed [2]. Indeed, existing methods have some problems in computation time and/or accuracy and thus other approaches should also be studied.

Recently, *support vector machines* (SVMs) have been applied to classification of protein structures, where SVMs are a statistical method widely used in bioinformatics and other various areas [6,15]. In order to apply SVMs to protein structures, a *kernel function* or a *feature vector* for protein structure is required. Dobson and Doig developed a feature vector based on various information on proteins [7], which includes secondary-structure content, amino acid propensities, surface properties and ligands. Their feature vector was applied to classification of proteins into enzymes and non-enzymes. Borgwardt *et al.* developed kernel functions based on graph kernels [4], where each protein structure is represented

T. Shibuya et al. (Eds.): PRIB 2012, LNBI 7632, pp. 233–242, 2012.

as a graph using secondary structure information. In order to improve the prediction accuracy, they also used additional features similar to those used by Dobson and Doig. Qiu *et al.* proposed a kernel for protein structures using a structure alignment algorithm [14]. Though these methods are very useful for predictions, it is difficult to extract structural information or common regions of proteins from the results of SVM learning. Therefore, it is desirable to develop a method with which structural information and/or common regions can be extracted.

In this paper, we propose a simple feature vector for finding common regions of protein structures. The proposed feature vector is based on the concept of *spectrum kernel* for sequence data [11]. The spectrum kernel uses a feature vector based on the numbers of occurrences of substrings of fixed length, where the length is usually short (e.g., 2 or 3). Though it is very simple, this method or similar methods are effectively applied to various problems in bioinformatics. Instead of substrings, our proposed feature vector uses a set of *template fragments of protein backbone structures*. And then, occurrences of similar fragments are taken into account in the feature vector. Different from the spectrum kernel, we use longer fragments each of which consists of several tens of Cα atoms. Moreover, similarities between fragments are measured by means of structural alignment because gaps cannot be ignored for such long fragments. For computing structural alignment, STRALIGN is employed, which was previously developed by one of the authors [1]. One of the important points of the proposed feature vector is that, different from existing methods [4,7], it uses structural information only and does not use any additional information such as secondary-structure content, amino acid propensities and so on.

The proposed feature vector is combined with SVMs in order to classify protein structures. Furthermore, it is combined with a feature selection method so as to find fragments conserved in multiple protein structures. To examine the proposed method, we performed computational experiments. The results suggest that the proposed method is useful to find common structural fragments in similar proteins.

2 Preliminaries

In this section, we briefly review SVM [6,15] and STRALIGN [1].

2.1 Support Vector Machine and Feature Vector

SVM is a kind of statistical learning method and is basically used for binary classification. Let POS and NEG be the sets of *positive examples* and *negative examples* in a training data set, where each example is represented as a point in d-dimensional Euclidean space (see Fig. 1). Then, an SVM finds a hyperplane h such that the distance between h and the closest point is the maximum (i.e., the margin is maximized) under the condition that all points in POS lie above h, and all points in NEG lie below h. Once this h is obtained, we can infer that a new test data is positive (resp. negative) if it lies above h (resp. below h).

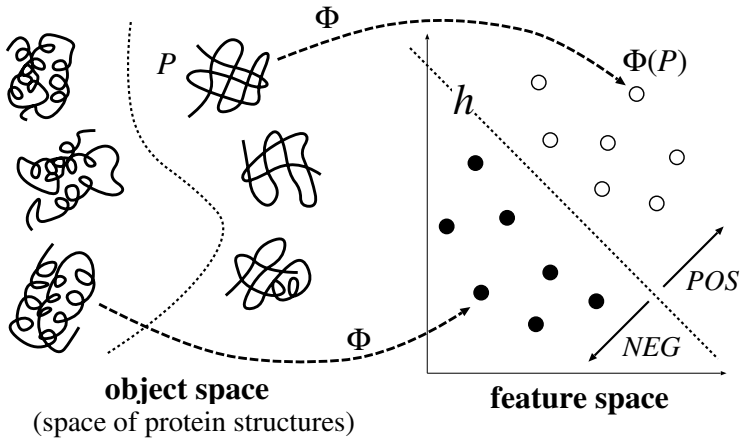

Fig. 1. Support vector classification (left) and feature map (right). In order to apply SVMs to analysis of protein structures, each structure should be mapped to a point (feature vector) in feature space.

If there does not exist h that completely separates POS from NEG, the SVM finds h which maximizes the *soft margin*, where we omit details of the soft margin [6,15].

In order to apply SVMs to real-world problems, it is important to design a *feature vector* or a *kernel function* suited to an application problem since objects to be classified are not usually points in Euclidean space. That is, we should find a feature mapping Φ from the object space $\mathcal{X}$ to the d-dimensional Euclidean space $\mathcal{R}^d$ (we can even consider infinite dimensional space). Then, $\Phi(x)$ is called a *feature vector* and $\mathcal{R}^d$ is called the *feature space*. That is, Φ transforms an object $x \in \mathcal{X}$ to a feature vector $\Phi(x)$:

$$x \in \mathcal{X} \Longrightarrow \Phi(x) \in R^d.$$

We also define a kernel K from $\mathcal{X} \times \mathcal{X}$ to $\mathcal{R}$ by

$$K(x, y) = \Phi(x) \cdot \Phi(y),$$

where $\Phi(x) \cdot \Phi(y)$ is the inner product between vectors $\Phi(x)$ and $\Phi(y)$. $K(x, y)$ is regarded as a measure of similarity between x and y.

2.2 STRALIGN

Protein structure alignment is a problem of finding amino acid pairs occupying spatially equivalent positions, given two 3D protein structures. Though the output of protein structure alignment is almost the same as that of pairwise sequence alignment, structural similarities are considered instead of similarities of

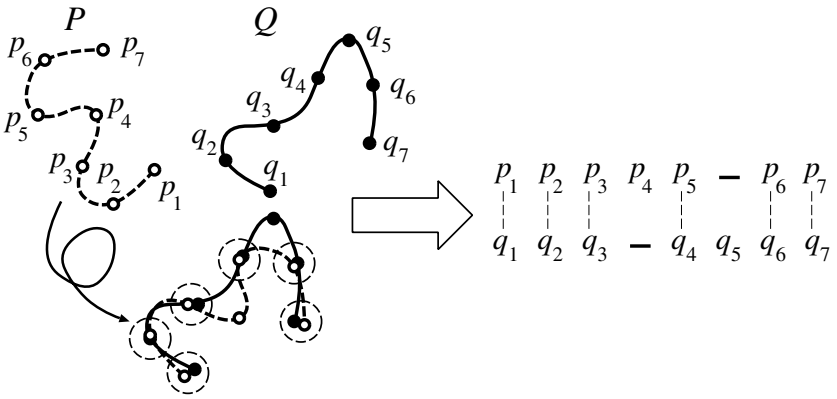

Fig. 2. Structure alignment is obtained by computing optimal superposition of two structures

amino acids (see Fig. 2). While many methods have been proposed for structure alignment [9,16], we use STRALIGN developed by one of the authors [1] because it is designed based on concrete theoretical foundation and is easy to modify. STRALIGN computes structure alignment in the following way.

STEP 1: A series of initial superpositions are computed from pairs of structural fragments (of length 10-20) using a standard technique to compute an optimal superposition without gaps (i.e., RMS (root mean squares) fitting).
STEP 2: For each of such superpositions, a rough alignment is first computed using a dynamic programming technique, and then is refined through an iterative improvement procedure which also uses dynamic programming.
STEP 3: Finally, the best alignment among those is selected as an output.

3 Method

In the proposed method, each protein structure P in training and test data sets is transformed into a feature vector $\Phi(P)$ and then SVM learning and classification are performed in a usual manner. Furthermore, feature selection is performed in order to extract common structural fragments. In the following, we describe outlines of computation of a feature vector and selection of important features.

3.1 Feature Vector Based on Similarity of Structural Fragments

In this work, each protein structure is represented by a sequence of positions of Cα atoms. Let $P = (p_1, p_2, \ldots, p_n)$ be a sequence of positions of Cα atoms. In the proposed method, a feature vector $\Phi(P)$ for protein structure P is defined as follows (see also Fig. 3).

Let L be the length of a structural fragment, where a fragment is a consecutive sequence of positions of Cα atoms, and $L = 40$ was employed in this work based on several trails. Let $\mathcal{T}$ be a set of template structures. Let $Q = (q_1, \ldots, q_m)$ be a template structure in $\mathcal{T}$. A set of fragments $frag(Q)$ from Q is defined by

$$frag(Q) \ = \ \{ \ (q_{i\Delta+1}, q_{\ i\Delta+2}, \ldots, q_{i\Delta+L}) \mid i = 0, 1, 2, \cdots \text{ and } i\Delta + L \leq m\},$$

where $\Delta = 10$ was used in this work. Then, a set of template fragments $\mathcal{F}$ is defined as

$$\mathcal{F} = \bigcup_{Q \in \mathcal{T}} frag(Q).$$

That is, a set of template fragments contains several fragments from each template structure, where template structures are selected from positive and negative classes (but not included in training or test data set).

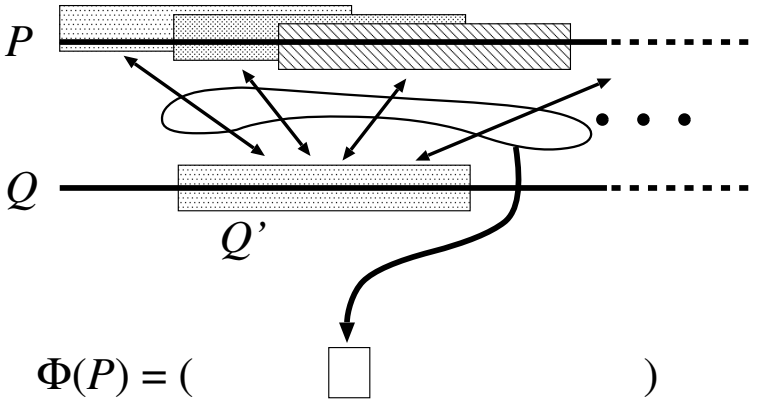

Fig. 3. Computation of a feature vector. Each coordinate in a feature vector corresponds to template fragment Q', where the coordinate value is defined by the sum of the scores for fragments in P against Q'.

For a structural fragment P' from a training or test protein structure P and a template fragment Q', we define the score $w(P', Q')$ by

$$w(P', Q') = \frac{\text{the number of superposed residue pairs}}{|P|},$$

where $|P|$ denotes the number of residues in P. We used this measure to evaluate the result of structural alignment between P' and Q' because STRALIGN tries to maximize the number of superposed residue pairs within some distance threshold. Then, the feature vector $\Phi(P)$ for a training or test protein structure P is defined by

$$\Phi(P) \ = \ \left(\sum_{P' \in frag(P)} w(P', Q') \right)_{Q' \in \mathcal{F}}.$$

That is, each coordinate value corresponding to a template fragment $Q' \in \mathcal{F}$ is defined by the sum of the scores for fragments of P against Q'.

3.2 Feature Selection

In order to find conserved structural fragments, we employ Recursive Feature Elimination (RFE) [8], which is a well-known feature selection method for SVMs. Different from the original RFE [8], we use the prediction accuracy (for the training data set) as a measure for eliminating features. Moreover, pre-processing based on Pearson correlation coefficient is introduced so as to eliminate redundant features efficiently. The following is an outline of our feature selection method:

STEP 1: Let $\mathcal{F}_0$ be a set of all template fragments.
STEP 2: Compute Pearson correlation coefficient between each $f \in \mathcal{F}$ and the class (i.e., positive or negative).
STEP 3: Let $\mathcal{F}$ be the subset of $\mathcal{F}_0$ consisting of fragments with H highest coefficients ($H = 30$ in this work).
STEP 4: For all $Q' \in \mathcal{F}$, perform SVM training using $\mathcal{F} - \{Q'\}$.
STEP 5: Let Q'' be the feature such that the classification accuracy for $\mathcal{F} - \{Q''\}$ is the highest.
STEP 6: Let $\mathcal{F} \leftarrow \mathcal{F} - \{Q''\}$.
STEP 7: Repeat STEPS 4-6 until reaching the specified number of features K.

It should be noted that $H = 30$ and $K = 3$ were used in this work.

4 Computational Experiments

We performed preliminary computational experiments in order to examine the potential power of the proposed method. We used protein structure data from ASTRAL [5] and SCOP [3] databases, where these two databases are closely related. We used the structure and classification data of ASTRAL SCOP 1.69 with less than 40% sequence identity. All experiments were performed on a PC cluster with AMD Opteron Model 280 (2.4GHz) CPUs, where each evaluation (i.e., combination of fold class and the feature selection method) took several minutes using one CPU in most cases.

First, we examined the accuracy of binary classification using SVM and feature vector $\Phi(P)$, where we employed SVM^{light} [10] for SVM learning and classification. We used the following eight SCOP fold classes each of which contains sufficient number of non-homologous proteins:

a.24: Four-helical up-and-down bundle,
a.118: Alpha-alpha superhelix,
b.29: Concanavalin A-like lectins/glucanases
b.40: OB-fold,
c.1: TIM beta/alpha-barrel,

c.23: Flavodoxin-like,
d.15: Beta-Grasp (ubiquitin-like),
d.58: Ferredoxin-like.

For each fold class F, we examined binary classification (i.e., predict whether or not a given protein structure belongs to F). For that purpose, 40 protein structures were randomly selected from F as positive data and 40 protein structures were randomly selected from other fold classes as negative data. As for template structures, 4 protein structures were randomly selected from F and 4 protein structures were randomly selected from other fold classes, under the condition that these 8 structures were different from the above 80 protein structures. Then, $\Phi(P)$ was computed for each P of 80 protein structures. Using these feature vectors and SVM^{light}, 5-fold cross validation was performed. For each fold class, *sensitivity*, *specificity* and *overall accuracy* are shown in the left part (corresponding to "all features") of Table 1. It should be noted that these measures are defined by:

$$\text{sensitivity} = \frac{TP}{TP+FN},$$
$$\text{specificity} = \frac{TP}{TP+FP},$$
$$\text{accuracy} = \frac{TP+TN}{TP+FP+TN+FN},$$

where TP, FP, TN and FN denote the numbers of true positives (structures correctly classified to the target fold class), false positives, true negatives and false negatives, respectively. It is seen that overall accuracies are reasonably good though these may not be the best among existing methods. It should be noted that we do not pay much attention to optimization of classification accuracy in this paper. Instead, we are more interested in identification of conserved fragments.

Next, we applied the proposed feature selection method to the same data sets, where the target number of features (i.e., the number of structural fragments) was set to 3 (i.e., $K = 3$). For comparison, we examined a very simple method that selects features with 3 highest Pearson correlation coefficients. Then, 5-fold cross validation was performed for each case and the results are shown in the middle and right parts of Table 1. It is very interesting to note that feature selection is useful to increase the classification accuracy. The results suggest that protein structures are well-classified by using only a small number of fragments. It is also seen that the RFE-based selection method is better than or as good as the Pearson-based selection method in most cases. Thus, the proposed feature selection method is considered to be useful for selecting fragments for protein structure classification.

Finally, we measured the conservation ratio of the best fragment among 3 fragments selected by the proposed method. For each fold class and for each of positive and negative data sets, we calculated the ratio of the number of protein structures containing the fragment to the total number of protein structures

Table 1. Comparison of classification accuracy (%) for different sets of features. Bold numbers correspond to the best classification accuracies among the three methods.

	all features			REF-based selection			Pearson-based selection		
Class	Sens.	Spec.	Acc.	Sens.	Spec.	Acc	Sens.	Spec.	Acc
a.24	90.0	90.0	90.0	92.5	90.2	**91.3**	92.5	90.2	**91.3**
a.118	100.0	93.0	96.4	100.0	97.6	**98.8**	97.5	100.0	**98.8**
b.29	72.5	100.0	86.3	87.5	89.7	**88.8**	80.0	86.5	83.8
b.40	90.0	81.8	85.0	70.0	96.6	**83.8**	65.0	92.9	80.0
c.1	87.5	83.3	85.0	87.5	87.5	87.5	95.0	84.4	**88.8**
c.23	72.5	85.3	80.0	87.5	89.7	**88.8**	80.0	86.5	83.8
d.15	77.5	93.9	86.3	75.0	93.8	**85.0**	67.5	96.4	82.5
d.58	52.5	80.8	70.0	60.0	85.7	**75.0**	57.5	88.5	**75.0**

Table 2. Conservation ratios (%) of selected fragments

	Fold Class							
	a.24	a.118	b.29	b.40	c.1	c.23	d.15	d.58
Positive	85.0	92.5	91.4	42.5	80.0	70.0	62.5	72.5
Negative	30.0	10.0	28.6	0.0	10.0	20.0	7.5	40.0

(i.e., 40) in the data set, where protein structure P is regarded to contain fragment Q' if the number of superposed residue pairs is no less than $0.65 \cdot |Q'|$. It is to be noted that the ratio should be high for positive data whereas it should be low for negative data. The result is shown in Table 2, where the threshold of $0.8 \cdot |Q'|$ was used for the case of 'a.118' (since the ratios for positive/negative data sets were 100%/60% if the threshold of $0.65 \cdot |Q'|$ was used). It is observed that good conservation ratios were obtained for most cases. For the case of 'd.58', the ratios were not good. But, it is consistent with the classification result in Table 1. For the case of 'b.40', the ratio for positive data was low. However, the ratios were 65.0%/0% if the threshold of $0.5 \cdot |Q'|$ was used. In summary, the proposed feature selection method is considered to be useful for selecting conserved fragments. It should be noted that, though conservation of a single fragment was examined here, multiple fragments are required to obtain the results of Table 1 and thus selection of multiple features is still important.

5 Concluding Remarks

We proposed a method for finding conserved regions in similar proteins. The method is a combination of a new feature vector based on structure alignment for fragments with two techniques in statistical learning: support vector machines and feature selection. It should be noted that, different from a common approach to identify conserved regions, the proposed method does not use multiple structure alignment though it uses pairwise structure alignment for fragments. The results of preliminary computational experiments suggest that the proposed method is useful to identify important structural fragments.

One of important future work is to perform rigorous and larger scale computational experiments, which include (i) adjustment of parameters (e.g., L, Δ, K and H) used in the method, (ii) study of the sensitivity of these parameters, (iii) comparison with other kernels for protein structures (e.g., [4,7,14]), and (iv) examination of other feature selection methods. It is also important to study biological meaning and/or significance of the selected fragments.

In the proposed method, configurations between fragments are not taken into account. However, configurations between fragments may play an important role in protein functions. In particular, such information seems important if we would like to predict interactions between proteins and/or interactions between proteins and chemical compounds. Therefore, a feature vector and/or a kernel function reflecting such information should also be developed.

References

1. Akutsu, T.: Protein structure alignment using dynamic programming and iterative improvement. IEICE Trans. Inf. Syst. E79-D, 1629–1636 (1996)
2. Akutsu, T., Halldórsson, M.M.: On the approximation of largest common subtrees and largest common point sets. Theoret. Comp. Sci. 233, 33–50 (2000)
3. Andreeva, A., Howorth, D., Brenner, S.E., Hubbard, T.J.P., Chothia, C., Murzin, A.G.: SCOP database in 2004: refinements integrate structure and sequence family data. Nucleic Acids Res. 32, D226–D229 (2004)
4. Borgwardt, K.M., Ong, C.S., Schönauer, S., Vishwanathan, S.V.N., Smola, A.J., Kriegel, H-P.: Protein function prediction via graph kernels. Bioinformatics 21, i47–i56 (2005)
5. Chandonia, J.M., Hon, G., Walker, N.S., Lo Conte, L., Koehl, P., Levitt, M., Brenner, S.E.: The ASTRAL compendium in 2004. Nucleic Acids Res. 32, D189–D192 (2004)
6. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20, 273–297 (1995)
7. Dobson, P.D., Doig, A.J.: Distinguishing enzyme structures from non-enzymes without alignment. J. Mol. Biol. 330, 771–783 (2003)
8. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46, 389–422 (2002)
9. Holm, L., Sander, C.: Protein structure comparison by alignment of distance matrices. J. Mol. Biol. 233, 123–138 (1993)
10. Joachims, T.: Making large-scale SVM learning practical. In: Schölkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods - Support Vector Learning, pp. 41–56. MIT Press (1999)
11. Leslie, C., Eskin, E., Noble, W.S.: The spectrum kernel: a string kernel for SVM protein classification. In: Proc. Pacific Symposium on Biocomputing, vol. 7, pp. 564–575 (2002)
12. Lupyan, D., Leo-Macias, A., Ortiz, A.R.: A new progressive-iterative algorithm for multiple structure alignment. Bioinformatics 21, 3255–3263 (2005)
13. Pearl, F.M., Bennett, C.F., Bray, J.E., Harrison, A.P., Martin, N., Shepherd, A., Sillitoe, I., Thornton, J., Orengo, C.A.: The CATH database: an extended protein family resource for structural and functional genomics. Nucleic Acids Res. 31, 452–455 (2003)

14. Qiu, J., Ben-Hur, A., Vert, J.-P., Noble, W.S.: A structural alignment kernel for protein structures. Bioinformatics 23, 1090–1098 (2007)
15. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge Univ. Press (2004)
16. Shindyalov, I.N., Bourne, P.E.: Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. Protein Eng. 11, 739–747 (1998)
17. Ye, Y., Godzik, A.: Multiple flexible structure alignment using partial order graphs. Bioinformatics 21, 2362–2369 (2005)