Discovering Sequence-Structure Motifs from Protein Segments and Two Applications

T. Tang, J. Xu, and M. Li

Pacific Symposium on Biocomputing 10:370-381(2005)

DISCOVERING SEQUENCE-STRUCTURE MOTIFS FROM PROTEIN SEGMENTS AND TWO APPLICATIONS

THOMAS TANG, JINBO XU, MING LI

School of Computer Science, University of Waterloo,
200 University Ave. W., Waterloo, Ont., N2L 3G1, Canada
{tcktang,j3xu,mli}@cs.uwaterloo.ca

We present a novel method for clustering short protein segments having strong sequence-structure correlations, and demonstrate that these clusters contain useful structural information via two applications. When applied to local tertiary structure prediction, we achieve ~60% accuracy with a novel dynamic programming algorithm. When applied to secondary structure prediction based on Support Vector Machines, we obtain a ~2% gain in Q_3 performance by incorporating cluster-derived data into training and classification. These encouraging results illustrate the great potential of using conserved local motifs to tackle protein structure predictions and possibly other important problems in biology.

1 Introduction

A major obstacle for protein tertiary structure prediction lies in the complexity of modeling protein 3D conformations due to the large degree of structural freedom and complicated interactions among residues. Previous models of computation include a number of lattice as well as off-lattice models [1]. A recently emerging model treats a protein as a composition of small local structural motifs, a concept inspired by the conjecture that a newly created polypeptide forms local folds in parts before settling to its final fold [2]. This model manages to reduce the size of protein conformational space to a point where many search-based prediction strategies finally become feasible. As a result, extraction of local motifs through classification of protein segments has always been a subject of intense study.

We initially created RAPTOR [3], an innovative protein tertiary structure predictor based on optimal threading by linear programming. This development has stimulated our interest in *ab initio* structural prediction and led us to investigate local fold information through clustering. The current results to be presented include a novel method for clustering protein segments with strong sequence-structure correlations, and two applications of the resultant clusters to structural predictions for demonstrating their usefulness.

2 Clustering of Short Protein Segments

Methods for clustering short protein segments are generally divided into two groups: a) those with clustering based on structure alone [4, 5], and b) those with

clustering based on both sequence and structure [6, 7]. Methods in the former omit sequence information, thus using the clusters they produce in *ab initio* structural prediction requires external guidance such as a global energy function. Since this study depends on sequence information to do prediction, we need a clustering method in the second group instead. Existing methods in this group perform clustering in two stages. Some of them first classify segments into clusters solely by sequence similarity and then sub-classify members in each cluster by structural similarity [6], while others did the reverse [7]. A problem associated with the two-stage approach is that segments with similar sequence patterns and folds might not as clearly reveal such a relationship when one looks at sequence and structure as separate entities. Those segments are likely to get misclassified in either or both stages. In this paper, we present a one-stage method, which will eliminate the deficiency by considering both sequence and structure together throughout the whole clustering process.

2.1 Segment Distance

For each residue i, its tertiary structure is represented by its *phi* (φ_i) and *psi* (ψ_i) angles in degrees, and its sequence information by frequency profiles comprising f_{ij} for amino acid j. Given segments x and y of length L, their distance $D(x, y)$ is:

$$D(x, y) = \begin{cases} \sqrt{\sum_{i=0}^{L-1} \left(\left(\frac{\Delta\varphi_i}{360} \right)^2 + \left(\frac{\Delta\psi_i}{360} \right)^2 + \sum_{j=0}^{19} \Delta f_{ij}^{\,2} \right)} & \text{if } \max(\Delta\varphi_i, \Delta\psi_i) \le \theta \ \ \forall\, i \\ \infty & \text{otherwise} \end{cases} \qquad (1)$$

Symbol Δ denotes the absolute difference in the associated quantity. Value θ restricts the largest dihedral angle difference permitted, and is L-dependent so as to allow higher leniency for longer segments. Eq. (1) has two ideal properties. First, it encompasses differences in both sequence patterns and structures, hence allowing one-stage clustering. Second, it is the Euclidean distance between two points so it satisfies the triangular inequality, a qualifying condition for use in clustering [4]. Note that the validity of Eq. (2) justifies the assumption that contributions from differences in structure and in sequence have equal weights.

$$0 \le \left(\frac{\Delta\varphi_i}{360} \right)^2 + \left(\frac{\Delta\psi_i}{360} \right)^2 \le 2 \text{ and } 0 \le \sum_{j=0}^{19} \Delta f_{ij}^{\,2} \le 2 \quad \forall\, i \in [0, L-1] \qquad (2)$$

2.2 Cluster Radius

Besides a distance function, we need a threshold, called *cluster radius*, to tell if two segments are sufficiently close to be grouped together. The choice of cluster

radius is crucial: being too small yields a handful of clusters capturing only the most conserved motifs, while being too large yields coarse clusters contaminated with irrelevant segments. A systematic way exists to determine a suitable radius for a given segment length. First, segments of that length are extracted from a large database of non-redundant proteins whose structures are known. An ideal database is PDB Select 25 [8]. The set of all segments are then divided in half, and distances between segments in different halves are computed. The resultant figures form a normal distribution with mean μ and standard deviation σ. The radius is set to $\mu - 3\sigma$, corresponding to a confidence interval of 99.73%. This choice of radius is found to consistently deliver clusters of reasonable quality.

2.3 Segment Preparation

Eq. (1) requires sequence profiles for both segments showing the frequency for each amino acid at each position. The profiles in this study are generated from multiple alignments in the HSSP database [9], and post-processed with the Voronoi Monte Carlo algorithm [10] to correct for unequal representations. Aside from profiles, secondary structure labels are also gathered, and for that the DSSP secondary structure labeling [11] is chosen due to its popularity.

2.4 Clustering Algorithm

The novel clustering algorithm is derived from the famous k-means algorithm [12], modified to allow a variable number of clusters [13]. It makes use of a special cluster called the *residue cluster* to hold segments failing to get classified due to their unique sequence patterns or shapes. The residue cluster is initially empty. Note that once a segment is placed into the residue cluster, it may no longer be used to start a new cluster. The algorithm is outlined below.

Protein Segment Clustering Algorithm

Input: cluster radius r, minimum size m, segment set S, maximum trial count t

1. Create empty residue cluster C_{res}
2. Repeat until no changes or t trials have been exhausted
3. For each segment $s \in S$ do
4. Find cluster closest to s, or set distance to ∞ if none exists yet
5. If distance $\leq r$ then move s to new cluster and update old cluster
6. Otherwise, if $s \notin C_{res}$ then create new cluster with s as centroid
7. Merge all nearby clusters (with distance $< 0.5r$)
8. For each cluster smaller than m do
9. Eliminate cluster and transfer all its segments to C_{res}
10. Return the final set of clusters

2.5 Experiments and Results

This section presents results on clustering a set of 396 non-redundant protein peptides referred to as CB396 [14]. Segment length L was set to 8, a value small enough to allow clusters of reasonable size but large enough to capture local residue interactions. In fact, it has been shown that segments of length 8 are very effective at preserving local sequence-dependent information [2]. The cluster radius was set to 1.2 based on the method described in Section 2.2. Both the minimum cluster size and maximum trial count were set to 5. Value of symbol θ in Eq. (1) was set to 120^o, a reasonable limit for length-8 segments [6].

The output comprised 357 clusters, but the number of distinct structural motifs was much less since many clusters either had the same fold, or were overlapping images of the same motif. For instance, 89 clusters were helices, showing the motif's abundance and its variety in sequence patterns. In summary, all motifs in the I-sites library [6] had been discovered together with some new ones. Examples of new motifs are shown in Figure 1. The motif in Figure 1(a) is characterized by a strong preference for hydrophobic residues at position 3 followed by a strong preference against them at the next position, indicating a possible emergence from inside the protein to the surface. The motif in Figure 1(b) is characterized by a GLY at position 3, a conserved hydrophobic residue at position 4, and finally an ASN or ASP at position 5. Descriptions for the motif in Figure 1(c) and the rest are omitted due to space limitation.

3 *Ab Initio* Local Tertiary Structure Prediction

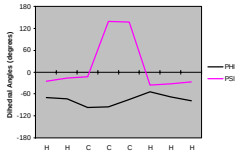
The first application of motif clusters is aimed at the *ab initio* prediction of local tertiary structures – the prediction of tertiary structures of short protein segments based solely on the sequence information contained in the segments. Success in resolving local structure prediction will be a major milestone in fold recognition, homology detection, and understanding of the protein folding process.

3.1 Assigning Clusters to Protein Segments

Scoring function $K_c(s)$, shown in Eq. (3), computes the likelihood of a length-L segment s belonging to cluster c based on sequence composition. Symbols s_{ij} and c_{ij} denote the frequency of amino acid j at position i on s and c's centroid respectively. Symbol b_j denotes the background frequency for amino acid j.

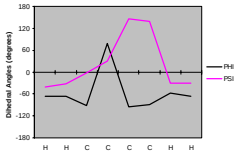
$$K_c(s) = \log_2 \left(\frac{\prod_{i=0}^{L-1} \sum_{j=0}^{19} s_{ij}\, c_{ij}}{\prod_{i=0}^{L-1} \sum_{j=0}^{19} s_{ij}\, b_j} \right) \tag{3}$$

a)



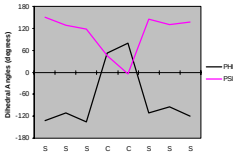
	0	1	2	3	4	5	6	7
A	·	1	·	·	-1	·	·	·
R	-1	1	·	-1	-1	·	-1	·
N	-1	·	·	·	1	·	·	·
D	-1	1	·	·	2	·	1	1
C	·	-3	·	·	-1	-3	-2	·
Q	·	·	·	·	·	·	1	1
E	·	1	·	-1	·	·	1	1
G	-1	-1	-1	-1	·	-1	-1	-1
H	-2	·	·	·	·	-1	-1	·
I	·	-1	·	1	-2	·	-2	-2
L	1	-1	·	1	-1	·	-1	-1
K	·	1	·	-1	·	·	·	·
M	1	·	1	1	-1	·	-2	-1
F	·	-1	·	·	-2	-1	-2	-1
P	·	·	-2	-2	2	1	·	-1
S	·	·	·	·	1	·	·	·
T	·	·	·	·	·	·	·	1
W	1	-2	·	-1	-2	-1	-2	·
Y	·	-1	·	·	-1	·	-1	·
V	·	-1	·	1	-1	-1	-1	·

b)



	0	1	2	3	4	5	6	7
A	·	1	1	-1	-1	-1	·	·
R	1	·	·	-1	·	-1	·	1
N	·	·	·	1	-2	1	·	·
D	·	·	-1	·	-2	2	-1	1
C	-2	-2	-2	-2	·	-2	-1	-1
Q	·	1	·	·	·	-2	-1	·
E	1	1	·	-1	-1	·	·	1
G	-2	-1	-1	3	-2	·	-1	-1
H	·	·	1	-1	-1	-1	·	·
I	-1	-1	-1	-3	1	-3	·	-2
L	-1	-1	1	-2	1	-2	1	-1
K	2	1	·	·	·	·	·	1
M	-1	-1	1	-1	1	-2	·	-1
F	-2	-2	1	-2	1	-1	1	-2
P	-2	-2	-4	-2	·	1	2	·
S	·	·	-1	-1	-2	1	-1	·
T	-1	·	·	-3	-1	1	-1	-1
W	-1	·	-1	-1	1	-1	·	-1
Y	-1	-2	1	-2	1	-1	·	-1
V	-1	-1	-2	-3	1	-2	·	-1

c)



	0	1	2	3	4	5	6	7
A	·	·	-1	-1	-1	·	-1	-1
R	·	·	·	·	·	1	-1	1
N	-1	·	·	2	1	1	-1	-1
D	-2	1	1	2	1	·	-1	-1
C	·	-1	-1	-4	-3	-3	-1	1
Q	·	1	·	·	-1	1	·	·
E	-1	·	·	·	·	1	·	·
G	-2	-1	·	1	2	·	-1	-2
H	-1	·	·	-1	-1	·	-2	·
I	1	·	·	-3	-2	-2	1	1
L	·	·	·	-3	-2	-2	1	·
K	-1	·	·	1	·	2	·	·
M	·	·	·	-2	-3	-1	·	-1
F	1	·	·	-2	-2	-1	·	1
P	-1	-2	-1	-1	-1	-1	·	-2
S	-1	·	·	·	·	·	-1	-1
T	1	·	·	-1	-1	·	1	·
W	-2	1	2	-4	-2	-2	1	-1
Y	1	·	1	-2	-1	·	·	1
V	1	1	·	-3	-1	·	1	1

Figure 1. Structural information and log-odds profiles for three novel motifs not listed in the I-sites database. Dot (·) represents background frequency.

A *cluster assignment*, or *assignment*, refers to an instance when a cluster is assigned to a segment based on a score computed via Eq. (3). The assignment is said to "cover" the segment and its residues. Each assignment has three basic attributes: the cluster being assigned, the segment being covered, and the score associated with the pair. Finally, a *cluster assignment rank*, denoted by *R*, means that each segment is assigned the *R* clusters yielding the *R* highest scores. The highest scoring assignment is at rank 1, the second highest at rank 2, etc.

3.2 Evaluating Local Structure Prediction

The evaluation scheme for local tertiary structure prediction was invented by Lesk [15]. It takes two parameters, a window size *w* and a RMSD threshold *t*. Given a true structure and its prediction, the scheme computes the percentage of residues found in length-*w* segments whose predicted structures are within *t* from the true structure after superposition. In this study, we use the same settings as Bystroff and Baker [6] to facilitate comparison (i.e. $w = 8$, $t = 1.4$ Å).

3.3 Eliminating Noise Clusters

In a large cluster set, some weak clusters capturing rare motifs possess similar sequence profiles as do the significant clusters capturing more common motifs. Those weak clusters tend to compete with the significant clusters for sequence similarity with target segments during cluster assignment, degrading prediction accuracy. Since they create "noise" that disturbs prediction, those weak clusters are called *noise clusters* and should be eliminated.

Clusters produced by the algorithm described in Section 2.4 are of minimum size *m*. If *m* is set too small, many noise clusters result. If it is set too large, significant clusters are lost. To determine *m* maximizing the predictive power for a set of clusters, the following empirical method is used:

Noise Cluster Elimination

Input: cluster set *C*, protein set *P*, minimum size bound $[m_l, m_h]$

1. For each *m* in range $[m_l, m_h]$
2. Remove clusters of size less than *m* from *C* to get *C'*
3. Get average prediction accuracy for *P* using *C'* as follows:
4. For each protein $p \in P$ do
5. Assign highest scoring cluster to each overlapping segment in *p*
6. Sort all assignments by score
7. Assign structures to *p* from highest scoring assignments
8. Evaluate prediction as described in Section 3.2
9. Return *m* and *C'* resulting in highest average prediction accuracy

Figure 2 shows the fluctuation in prediction accuracy as *m* increased from 5 to 25 inclusive. While the accuracy remained rather constant in the middle stretch, it rose and fell sharply at both ends. Prediction was compromised by the presence of noise clusters for small *m* (< 8) and the absence of significant clusters for large *m* (> 20). The optimal minimum cluster size was *m* = 16, yielding a prediction accuracy of 54.66%.

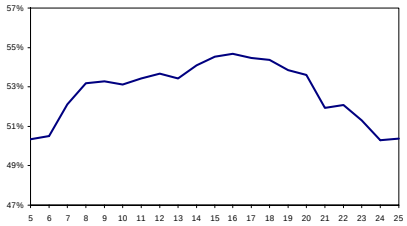


Figure 2. Fluctuation in prediction accuracy as minimum cluster size *m* increases from 5 to 25.

3.4 Improving Cluster Likelihood Function

Eq. (3) has been extended with the addition of a new term, as shown below:

$$K_c(s) = \log_2 \left(\frac{\prod_{i=0}^{L-1} \sum_{j=0}^{19} s_{ij}\, c_{ij}}{\prod_{i=0}^{L-1} \sum_{j=0}^{19} s_{ij}\, b_j} \right) - base_c \qquad (4)$$

The new term $base_c$ represents the basal cutoff score specific to cluster *c*. The old version (i.e. Eq. (3)) assumes a cutoff of 0 for all clusters, an intuitive choice for log-odds but nonetheless a crude assumption. The derivation procedure of cluster-specific cutoffs is based on a simple observation. Clusters capturing rare motifs are likely to introduce false positive (F^+) assignments, thus requiring a higher cutoff to avoid high a F^+ rate. On the other hand, clusters capturing common motifs are likely to introduce false negative (F^-) assignments and a lower cutoff is needed to suppress the F^- rate. The actual procedure for deriving the cutoffs is relatively unimportant and hence omitted due to space limitation.

3.5 Predicting Local Structure using Dynamic Programming (DP)

The algorithm to be presented is inspired by two observations. First, the cluster assignment most appropriately capturing the shape of a segment might not be the optimal (i.e. highest-scoring) one but a sub-optimal one. Second, if overlapping assignments have serious structural conflicts, then they should not be adopted together. We propose here an objective function for measuring the quality of a set of assignments when used together to form a prediction, and a DP approach for maximizing it. The objective function for a target protein *p* of length *n* is:

$$F(X) = q \sum_{i=0}^{n-1} \text{score}(X, i) - \sum_{i=0}^{n-1} \text{conflict}(X, i) \qquad (5)$$

Function $F(X)$ returns the objective score for a set of cluster assignments X. Symbol q is a non-negative constant for balancing the two parts representing the total score and conflict induced by X. It is set to 70 in this study, a value found empirically to yield one of the best predictions. Function score(X, i) is:

$$\text{score}(X, i) = \begin{cases} \text{score of highest scoring assignment in } X \text{ covering residue } i, \text{ or} \\ 0 \text{ if no assignment in } X \text{ covers residue } i \end{cases}$$

And function conflict(X, i) is:

$$\text{conflict}(X, i) = \begin{cases} \overline{\Delta\varphi} + \overline{\Delta\psi} \text{ between all pairs of assignments in } X \\ \quad \text{at positions covering residue } i, \text{ or} \\ 0 \text{ if at most 1 assignment in } X \text{ covers residue } i \end{cases}$$

Symbols $\overline{\Delta\varphi}$ and $\overline{\Delta\psi}$ denote the mean absolute difference in *phi* and *psi* angles respectively. Let L be the segment length and R be the assignment rank. To setup for the DP algorithm, the R highest-scoring assignments are made to each overlapping length-L segment along p. Let a_{ir} denote the assignment at rank r starting at position i, where $1 \le r \le R$ and $0 \le i \le n–L$. Define $A_i = \{a_{ir} \forall\, r\}$ and

$A = \{a_{ir}\}$. The algorithm is to find an assignment set $X^* \subseteq A$ such that X^* covers all residues in p and is optimal (i.e. maximizing objective function F).

Each assignment set is built progressively starting from the first residue by appending to the end one adjoining assignment at a time. Note that simply extending the current optimal set by adding to its tail the best available adjoining assignment does not guarantee optimality for the resultant set. The assignment just added may overlap with existing assignments in the set, introducing new conflicts that must be fixed by replacing those assignments, which in turn may cause more new conflicts with their prior overlapping assignments and necessitate further replacements. To avoid such propagation of conflicts, a more involved DP algorithm is needed.

When an assignment $\alpha \in A_i$ is appended to the end of assignment set X, it would come in contact with one or more trailing assignments in X. The arrangement of these trailing assignments and their ranks collectively form the *tail configuration* for X with respect to $\alpha \in A_i$, denoted by $tail_i(X)$. We define $tail_i(X)$ to be an empty tail configuration if X is too short to reach any assignment in A_i. For formulation purposes, we allow $tail_j(X)$ for $j > n–L$, in which case it is treated as if A_j actually existed.

For each position i starting from zero, the algorithm computes V_i, the collection of all optimal assignment sets X each having a unique non-empty $tail_{i+1}(X)$. The following recurrence ensures the optimality for each $V_{(i+1)t'}$, and the uniqueness and non-emptiness of the associated tail configuration t', so the inductive hypothesis holds for position i+1. Finally, we assign dihedral angles to residues in p by back-tracking the creation of X^*.

DP Recurrence for Local Tertiary Structure Prediction

Initial condition:

$V_0 = \{\{\alpha\} \; \forall \; \alpha \in A_0\}$

Inductive hypothesis for position i, $0 \le i \le n - L$:

$V_i = \{$All optimal assignment sets w each having a unique non-empty $tail_{i+1}(w)\}$

Recurrence:

Let $V_i' = \{w \cup \{\alpha\} \; \forall \; w \in V_i$ and $\alpha \in A_{i+1}\}$

For each unique non-empty tail configuration t'

$V_{(i+1)t'} = X \in \{w \in V_i \cup V_i' \mid tail_{i+2}(w) = t'\}$ s.t. $F(X)$ is maximized

$V_{i+1} = \{V_{(i+1)t'}\}$

Final solution:

$X^* = X \in \{w \in V_{n-L} \mid w$ has an assignment in $A_{n-L}\}$ s.t. $F(X)$ is maximized

Note that $|V_i|$ is bounded by $(R+1)^L - 1$, the number of all possible unique non-empty tail configurations. For each position i, the algorithm calculates the

objective value for $|V_i|*R$ new assignment sets, where each calculation is $O(L^3)$ if done carefully. Hence, the total runtime is $O(n\ |V_i|\ R\ L^3) = O(n\ L^3\ (R+1)^{L+1})$ for all n positions. Fortunately, typical values for R and L are small enough to make the runtime acceptable (e.g. $R = 3$ and $L = 8$ in this study).

3.6 Experiments and Results

A jackknife test was performed on CB396 [14], a set of 396 peptides selected through a very stringent procedure to ensure non-redundancy between members. The entire test consisted of 10 iterations, each of which involved splitting CB396 into two disjoint subsets in 80/20 ratio by residue count. The larger subset was then used for training and the smaller one for testing.

Note that testing sets containing more helices tend to yield higher accuracies than those containing more coils. Consequently, for results to be consistent, all testing sets should contain similar proportions of each secondary structure (SS). To guarantee such condition, the background proportion of each SS was first estimated from the whole CB396. Each repetition of the jackknife test then produced 50 pairs of training and testing sets, and used the pair whose testing set exhibited SS proportions most closely resembling the background ones.

Results for rank $R = 3$ are listed in Table 1. Overall, an average of 58.21% of all residues was found in a length-8 segment within 1.4 Å of the true structure, measured in RMSD. This is significant considering that the prediction relied solely on sequence information, without taking into account global forces such as disulfide bridges, hydrophobic effects, inter-group charges, and so on. The result is also a great improvement over that published by Bystroff and Baker [6], which was 50%. A breakdown in overall prediction accuracy by SS states reveals the real strengths and weaknesses of prediction using clusters. Helices were by far the most accurately predicted because they were the most conserved and abundant local motifs. Strands, albeit well conserved, were a lot harder to predict as their formation involved long-range residue interactions, something not captured by local motif clusters. Coils were the most difficult to predict since most of them lacked virtually any kind of detectable conserved patterns.

Table 1. Prediction accuracy of the DP algorithm obtained from a ten-iteration jackknife test on CB396 and evaluated using the scheme described in Section 3.2.

	Helix (%)	**Sheet (%)**	**Coil (%)**	**Total (%)**
1	86.22	44.91	40.58	58.51
2	84.54	39.61	40.52	56.19
3	84.27	44.71	41.07	58.03
4	84.65	43.22	40.19	57.45
5	86.12	43.63	42.87	59.15
6	88.11	42.92	43.05	59.69
7	84.83	44.35	40.99	57.76
8	84.94	43.98	42.90	58.83
9	86.09	43.22	42.78	58.86
10	83.68	45.30	40.61	57.62
Mean	*85.35*	*43.59*	*41.56*	*58.21*

4 Secondary Structure Prediction

The second application of clusters deals with enhancing secondary structure (SS) prediction. The target predictor [16] is the one based on Support Vector Machines (SVM) [17], so selected because it is one of the best available. As an overview, the procedure involves building a *Secondary Structure Confidence Profile* (SSCP) and using it as additional data for training and classification.

4.1 Secondary Structure Confidence Profile (SSCP)

The SSCP of a protein shows the confidence of each residue being in each of the three SS states, namely helix (H), strand (E), and coil (C). Given assignment rank R and target protein p, the method for creating SSCP first makes the R highest-scoring assignments to each length-L overlapping segment in p. Then, for each residue i and SS label $s \in \{H, E, C\}$, it computes $score_{is}$ by summing the scores of all assignments covering i with label s at the covering position. The value $score_{is}$ is then normalized to obtain ssc_{is}, the SS confidence for i belonging to state s. That is, $ssc_{is} = score_{is} / (score_{iH} + score_{iE} + score_{iC})$. The set of all ssc_{is} constitutes the SSCP for protein p.

4.2 Training of SVM Binary Classifiers

Fix a window half-width h such that each residue is represented by the sequence profile spanning $(2h + 1)$ columns, with the said residue in the middle. Each column is coded using 21 entries, where the extra entry is set when the window is extended beyond the ends of a protein. Together, each residue is coded by a total of $(2h + 1) * 21$ entries. When SSCP is incorporated into training, each column is coded with four additional entries. Each of the first three holds the SSCP confidence value for a different SS state, and the last is again set for the case when the window is extended beyond the ends of a protein. Hence, each residue is now coded by a total of $(2h + 1) * 25$ entries.

4.3 SVM Predictor Construction

Han and Sun [16] have demonstrated that different arrangements of SVM binary classifiers contribute to varying performance for the resultant SS predictor. One of the most effective configurations found in their study is called SVM MAX, which comprises three SVM binary classifiers, namely H/~H, E/~E, and C/~C. Each target residue is fed in parallel to all three classifiers, and assigned the SS label corresponding to the one giving the largest value. For optimal prediction, the half-width h for the three classifiers is set to 5, 4, and 3 respectively.

4.4 Experiments and Results

Three metrics were used to measure the quality of SS prediction. They were the Q_3, the Matthew's Correlation Coefficients (MCC) [18], and the Segment Overlap measure (SOV) [19]. The jackknife test and the target data set were as described in Section 3.6, except that each training set was also used to generate SSCP and train SVM in addition to creating motif clusters. Assignment rank *R* was set to 6. Parameters for SVM binary classifiers were 1.5 for error trade-off and 0.1 for γ in the radial basis function used as the kernel [16].

Table 2. Prediction accuracy of SVM MAX trained without SSCP (top values) and trained with SSCP (bottom values) in a ten-iteration jackknife test on CB396. A positive delta on the last row indicates an average improvement with SSCP (delta = mean bottom value – mean top value).

	Q_H (%)	**Q_E (%)**	**Q_C (%)**	**Q_3 (%)**	**C_H**	**C_E**	**C_C**	**SOV (%)**
1	74.30 76.34	55.44 60.90	78.49 78.60	72.15 74.10	0.63 0.68	0.51 0.55	0.53 0.55	68.57 70.63
2	79.03 78.85	57.33 63.39	79.07 79.86	74.44 75.97	0.66 0.69	0.56 0.59	0.57 0.58	69.77 71.58
3	80.36 81.57	50.26 55.77	76.87 78.58	71.94 74.35	0.64 0.69	0.51 0.56	0.53 0.55	69.50 70.15
4	76.37 76.92	52.78 58.07	77.18 77.99	71.61 73.29	0.62 0.65	0.51 0.54	0.53 0.54	68.37 70.52
5	77.21 79.13	52.96 58.54	78.32 78.10	72.36 74.23	0.65 0.68	0.51 0.55	0.54 0.55	70.23 71.43
6	79.72 81.89	56.68 61.96	77.05 77.79	73.57 75.84	0.65 0.71	0.55 0.59	0.55 0.57	71.58 73.01
7	78.23 79.25	52.07 55.06	77.62 78.65	72.09 73.56	0.64 0.67	0.51 0.53	0.53 0.55	70.11 71.61
8	76.23 77.50	51.23 56.07	76.56 77.80	70.81 72.86	0.62 0.66	0.50 0.53	0.51 0.53	66.54 68.57
9	76.25 77.82	56.36 62.51	79.52 79.33	73.13 75.01	0.64 0.68	0.55 0.59	0.55 0.56	70.78 72.92
10	74.93 77.24	50.48 56.39	77.72 77.72	70.70 72.86	0.61 0.65	0.49 0.53	0.52 0.54	66.34 67.97
Mean	*77.26* *78.65*	*53.56* *58.87*	*77.84* *78.44*	*72.28* *74.21*	*0.64* *0.68*	*0.52* *0.56*	*0.53* *0.55*	*69.18* *70.84*
Delta	*1.39*	*5.31*	*0.60*	*1.93*	*0.04*	*0.04*	*0.02*	*1.66*

The results are listed in Table 2. By combining SSCP with sequence profile for training and classification, SVM MAX predictor showed improvements in all Q_3, MCC and SOV measures. Specifically, SSCP contributed to an average Q_3 improvement of 1.93% by boosting the accuracy for helixes and strands, the latter in particular. In other words, SSCP helped the predictor be more certain when determining if a residue was part of a helix or strand. Moreover, the use of SSCP also resulted in visible improvements in all aspects of MCC and SOV. Unfortunately, improvements to Q_C and C_C were only minimal. After all, motif clusters could only identify regions with strong sequence-structure correlations, a

condition excluding most coils. Consequently, assignments made to segments in coil regions were mostly incorrect, producing unreliable SS confidence values.

5 Conclusion and Future Work

Motivated by the substantial improvement to secondary structure prediction [20], a repeat of this study using sequence profiles generated by PSI-BLAST [21] is currently underway. In the longer run, we are going to investigate another kind of sequence-structure motifs for capturing long-range inter-residue interactions. Our current sequence-structure motifs can only capture local interactions, so they are not very helpful for beta sheet prediction. Nevertheless, the partition of short protein segments into clusters of local sequence-structure motifs does have profound applications. These motif clusters achieve discretization of protein conformational space and provide an effective mapping between sequence and structure, all contributing to the success of their employment to both secondary and local tertiary structure prediction. The promising results obtained could mark the beginning of a wide range of potential applications for motif clusters, which include fold recognition, domain detection, and functional annotation.

References

1. X. Yuan, Y. Shao, C. Bystroff. *Comp Funct Genom*, 4(4):397-401, 2003
2. C. Bystroff, et al. *Curr Opin Biotechnol*, 7:417-421, 1996
3. J. Xu, M. Li, D. Kim, Y. Xu. *JBCB*, 1(1):95-117, 2003
4. R. Kolodny, P. Koehl, L. Guibas, M. Levitt. *J Mol Biol*, 323:297-307, 2002
5. C. G. Hunter, S. Subramaniam. *Proteins*, 50:580-588, 2003
6. C. Bystroff, D. Baker. *J Mol Biol*, 281:565-577, 1998
7. A. G. de Brevern, C. Etchebest, S. Hazout. *Proteins*, 41:271-287, 2000
8. U. Hobohom, M. Scharf, R. Schneider, C. Sander. *Protein*, 1:409-417, 1992
9. C. Sander, R. Schneider. *Proteins*, 9:56-68, 1991
10. P. Sibbald, P. Argos. *J Mol Biol*, 216:813-818, 1990
11. W. Kabach, C. Sander. *Biopolymers*, 22:2577-2637, 1983
12. B. S. Everitt. *Cluster Analysis 3rd Edition*, Halsted Press, New York, 1993
13. M. Anderbert. *Cluster analysis for applications*, Academic Press, NY, 1973
14. J. A. Cuff, G. J. Barton. *Proteins*, 34:508-519, 1999
15. A. M. Lesk. *Proteins Suppl*, 1:151-166, 1997
16. S. Hua, Z. Sun. *J Mol Biol*, 308:397-407, 2001
17. C. Burges. *Data Mining and Knowledge Discovery 2*, 121-167, 1998
18. B. W. Matthews. *Biochim Biophys Acta*, 405:442-451, 1975
19. A. Zemla, C. Venclovas, K. Fidelis, B. Rost. *Proteins*, 34:220-223, 1999
20. D. T. Jones. *J Mol Biol*, 292:195-202, 1999
21. S. F. Altschul et al. *Nucleic Acids Res*, 25(17):3389-402, 1997