BIOINFORMATICS

Vol. 20 no. 17 2004, pages 3099–3107
doi:10.1093/bioinformatics/bth370

Comparison of probabilistic combination methods for protein secondary structure prediction

Yan Liu[1,*], Jaime Carbonell[1], Judith Klein-Seetharaman[1,2] and Vanathi Gopalakrishnan[2]

[1]Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA15213, USA and [2]Center for Biomedical Informatics, University of Pittsburgh, Pittsburgh, PA15260, USA

Received on June 11, 2004; accepted on June, 12, 2004
Advance Access publication June 24, 2004

ABSTRACT

Motivation: Protein secondary structure prediction is an important step towards understanding how proteins fold in three dimensions. Recent analysis by information theory indicates that the correlation between neighboring secondary structures are much stronger than that of neighboring amino acids. In this article, we focus on the combination problem for sequences, i.e. combining the scores or assignments from single or multiple prediction systems under the constraint of a whole sequence, as a target for improvement in protein secondary structure prediction.

Results: We apply several graphical chain models to solve the combination problem and show that they are consistently more effective than the traditional window-based methods. In particular, conditional random fields (CRFs) moderately improve the predictions for helices and, more importantly, for beta sheets, which are the major bottleneck for protein secondary structure prediction.

Contact: yanliu@cs.cmu.edu

INTRODUCTION

Protein secondary structure prediction involves the projection of primary sequences onto a string of secondary structure assignments, such as helix, sheet or coil. It is widely believed that secondary structures can contribute valuable information to discerning how proteins fold in three dimensions.

Protein secondary structure prediction has been extensively studied for decades (Cuff and Barton, 1999; Rost, 2001). Recent improvements have been accomplished not only by incorporating evolutionary information, but also by combining the results of multiple independent prediction methods into a consensus prediction (Rost, 2001).

The architecture of a typical consensus prediction system is outlined in Figure 1. First, profile generation [(A) in Fig. 1], or feature extraction, converts the primary protein sequences to a set of features that can be used to predict the labels of secondary structures. Divergent profiles of multiple sequence alignments and a large variety of informative features have been used (Rost and Sander, 1993; Jones, 1999). Next, a sequence-to-structure mapping process [(B) in Fig. 1] outputs the predicted scores for each structure type using the features from (A) as input. Complex machine learning algorithms have been applied, including neural networks (Rost and Sander, 1993), recurrent neural networks (Pollastri *et al*., 2002), Support Vector Machines (SVMs) (Vapnik, 1995; Hua and Sun, 2001) and Hidden Markov Models (HMMs) (Bystroff *et al*., 2000). Then, the output scores from (B) are converted to secondary structure labels. This involves considering the influence of neighboring structures by structure-to-structure mapping (C) and physically removing unlikely conformations by a Jury system (D), also referred to as 'filters' or 'smoothers'. Some systems separate (C) and (D) for explicit evaluation while others keep them in one unit (Rost and Sander, 1993; King and Sternberg, 1996). Finally, a consensus is formed by combining predicted scores or labels from multiple independent systems into a single labeled sequence. Several methods have been applied to consensus formation, such as a complex combination of neural networks (Cuff and Barton, 2000), multivariate linear regression (Guermeur *et al*., 1999), decision trees (Selbig *et al*., 1999) and cascaded multiple classifiers (Ouali and King, 2000).

While profile generation (A) and sequence-to-structure mapping (B) have been studied extensively, the structure-to-structure mapping and jury system (C, D) have not been explored in detail although they are commonly used in various systems. Recent analysis by information theory also indicates that the correlation between neighboring secondary structures are much stronger than that of neighboring amino acids (Crooks and Brenner, 2004). From a machine learning perspective, both the jury system (C, D) and the

*To whom correspondence should be addressed

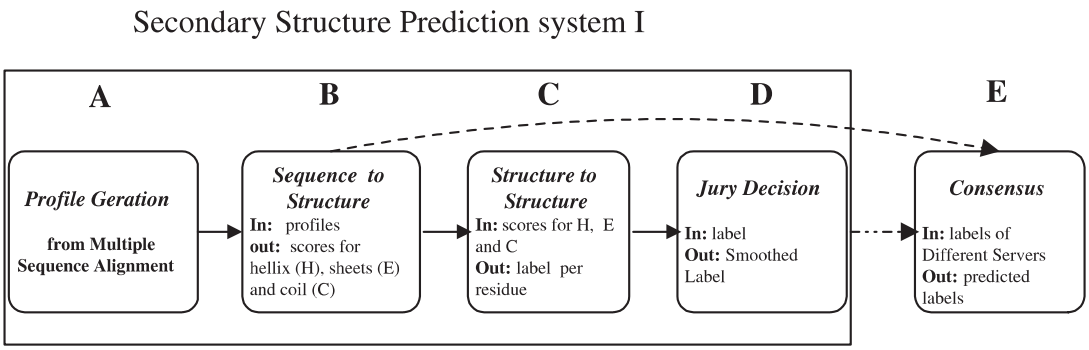

Fig. 1. The architecture of current secondary structure predictions [adapted from Rost and Sander (1993)].

consensus (E) can be formulated as the *combination problem for sequences*: given the predicted scores or labels, how should we combine them into the final labels, taking into account the dependencies of neighbors and constraints of a single protein sequence?

Note that the combination problem for sequences is distinct from another closely related task: given the predicted scores or labels from different systems for one residue, how can we combine them into the optimal labels? This task is a classical problem for machine learning known as an ensemble approach and many ensemble methods have been used for consensus formation. The difference between our task and the ensemble problem is that ensemble treats each residue as independent and does not consider the extra information from neighboring structures or constraints of a single sequence. Therefore, our combination problem is more general and difficult than a classical ensemble problem.

Previous methods for Jury and consensus use window-based approaches, i.e. taking predicted scores or labels from a sliding window and treating them as a classification problem (Rost and Sander, 1993; King and Sternberg, 1996; Sollich and Krogh, 1996; Krogh and Sollich, 1997; Selbig *et al.*, 1999; Cuff and Barton, 2000) (as shown in Fig. 2). However, the window-based methods cannot capture long-distance interactions, which are a hallmark of protein tertiary structures and known to influence the formulation and stability of secondary structures. Therefore, we propose the use of graphical chain models for the combination since they are able to consider the correlations between labels, to include long-distance interaction and to model the protein sequence as a whole.

MATERIALS AND METHODS

We formulate our combination problem as follows: given a protein sequence $P = x_1 x_2 \cdots x_N$, the raw output by a secondary structure prediction system is either a label sequence $L = l_1 l_2 \cdots l_N$, or a $N \times 3$ score matrix S, where $S_{ij} = S_j(x_i)$ is the score of residue x_i for class j, $j \in Y = \{H, E, C\}$ and $i \in \{1, 2, \ldots, N\}$. Taking the predicted labels L or score matrix S, we try to predict the true label $Y_1 Y_2 \cdots Y_N$.

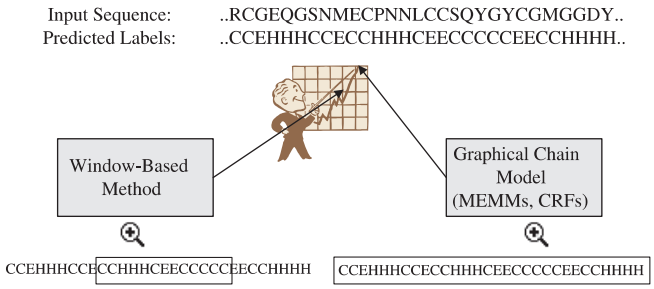

Fig. 2. Comparison of combination methods for protein secondary structure predictions.

Without loss of generality, we assume that (1) the predicted scores are non-negative and normalized; (2) for one residue x_i, the higher the score S_{ij}, the larger the probability that the residue x_i belongs to class j.

Traditional window-based combination

Window-based method for label combination The standard method for converting scores to predicted secondary structure labels is to assign the class with the highest score. After that, many systems employ rule-based methods to improve upon the first-pass assignment, i.e. the *label combination*, for instance: Rost and Sander manually define heuristic rules to remove helices with a length less than 3 and strands of length 1 (Rost and Sander, 1993; Salamov and Solovyev, 1995); King and Sternberg applied a decision tree algorithm to learn the rules automatically with 10-fold cross-validation (King and Sternberg, 1996).

Predefined heuristic rules, without considering the properties of the data, have not improved the accuracy consistently (Rost and Sander, 1993). In contrast, rules extracted automatically by supervised classifiers not only can generate the rules to filter out physically unrealistic predictions, but also can reduce the inductive biases from the particular learning algorithm that the system used for prediction, such as neural networks and SVMs (Wolpert, 1992).

The window-based label combination works as follows: given the labels predicted by a system $l_1 l_2 \cdots l_N$, and the

window size w, let $d = (w-1)/2$ be the half of the window size. The input features for residue x_i are the predicted labels within the window w, i.e. $\langle l_{i-d}, l_{i-d+1}, \ldots, l_{i+d-1}, l_{i+d}\rangle$ (a null label is assigned if the label does not exist). Then a rule-based classifier, such as decision tree or CART (Rost and Sander, 1993), can be applied. The window size w is a parameter with which we can tune the trade-off between including useful information and excluding 'noisy' more remote features.

Window-based method for score combination In current secondary structure prediction systems, *score combination* is used widely. Window-based score combination works similar to label combination except: (1) the input features for residue x_j are scores instead of labels, i.e. $[S_H(x_{i-d}), S_E(x_{i-d}), S_C(x_{i-d}), \ldots, S_H(x_{i+d}), S_E(x_{i+d}), S_C(x_{i+d})]$; (2) powerful classifiers, such as neural networks and k-Nearest-Neighbor, are used instead of rule-based classifiers.

Empirically, score combination has demonstrated more improvement in accuracy than label combination since the score $S_j(x_i)$ indicates the confidence of the prediction that residue x_i belong to class j and thus contains more information than a single label (Rost and Sander, 1993; Salamov and Solovyev, 1995; Jones, 1999; Guo *et al*., 2004). On the other hand, we can expect that applying label combination after score combination will hardly change the final predictions since the information from labels has been implicitly encoded in the scores (Rost and Sander, 1993). Both window-based label combination and score combination have the disadvantages of only considering the local information.

Graphical models for score combination

Simple graphical chain models, such as HMMs, have been successfully applied to secondary structure prediction (Karplus *et al*., 1998; Bystroff *et al*., 2000). HMMs are generative models that assume that the data are generated by a particular model. These models work by computing the joint distribution of observations $\mathbf{x}$ and states $\mathbf{y}$, $P(\mathbf{x}, \mathbf{y})$ and make predictions by using Bayes rules to calculate $P(\mathbf{y}|\mathbf{x})$. Two kinds of probability distributions are defined in HMMs: (1) the transition probabilities $P(y_i|y_{i-1})$ and (2) the observation probabilities $P(x_i|y_i)$. By the independence assumptions, i.e. $p(x_i|y_i) = p(x_i|y_i, y_{i-1})$, we have the joint probability $P(x_i, y_i|y_{i-1}) = P(x_i|y_i)P(y_i|y_{i-1})$. The graphical structure of HMMs is shown in Figure 3A.

Although successfully applied to many sequence data problems, HMMs are not appropriate for our combination task. First, it is difficult to include overlapping long-range features due to the independence assumption. Second, generative models such as HMMs, work well only when the underlying assumptions are appropriate. On the other hand, discriminative models do not make any assumptions and compute the posterior probability directly. Recently, the machine learning community has proposed several discriminative models for sequence data, such as Maximum Entropy Markov Models (MEMMs) (McCallum *et al*., 2000) and conditional random fields (CRFs) (Lafferty *et al*., 2001). They have been successfully applied to many applications including information retrieval and computer vision, and achieved significant improvement over HMMs (McCallum, 2003). Compared with window-based methods, these graphical models are able to take into consideration the correlations between labels and long-distance information. Therefore, we propose to use the discriminative graphical chain models for score combination. To the best of our knowledge, this approach has not been studied in previous protein secondary structure prediction literature and is the primary focus of our article.

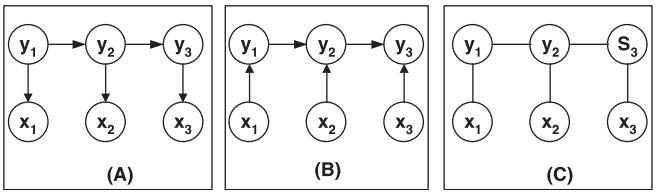

Fig. 3. Graphical structures of (**A**) simple HMM, (**B**) MEMM, (**C**) and chain-structured CRF.

Maximum entropy Markov models As shown in Figure 3B, MEMMs replace the generative joint probability $[P(\mathbf{x}, y_i|y_{i-1})]$ parameterization in HMMs with the conditional probabilities $P(y_i|y_{i-1}, \mathbf{x})$ based on an exponential model (McCallum *et al*., 2000):

$$P(y_i|y_{i-1}, \mathbf{x}) = \frac{1}{Z(y_{i-1}, \mathbf{x})} \exp\left[\sum_k \lambda_k f_k(\mathbf{x}, y_i, y_{i-1})\right], \quad (1)$$

where $Z(y_{i-1}, \mathbf{x})$ is a normalizing factor. The exponential models, derived by maximum entropy, are able to handle arbitrary, non-independent features, f_k, including long-distance interactions. The model parameter λ_k, i.e. the weight for feature f_k, is learned via maximizing the conditional likelihood of the training data $\prod_t P(\mathbf{y}_t|\mathbf{x}_t)$.

Despite the differences between HMMs and MEMMs, there is still an efficient dynamic programming solution to the problem of identifying the most likely state sequence *given* an observation. Compared to HMMs, McCallum *et al*. (2000) redefined $\alpha_i(y)$ to be the probability of being in state y at time i given the observation sequence up to time i. Then the recursive step is

$$\alpha_{i+1}(y) = \sum_{y' \in Y} \alpha_t(y') \cdot P(y|y', x_{i+1}). \quad (2)$$

Similarly, $\beta_i(y)$ is redefined to be the probability of starting from state y at time i *given* the observation sequence after

time i and the recursive step is

$$\beta_i(y') = \sum_{y \in Y} P(y|y', x_{i+1}) \cdot \beta_{i+1}(y).$$

Given the observation $x_1 x_2 \cdots x_N$, we can compute (1) the marginal mode of the optimal labels $l_1 l_2 \cdots l_N$ by

$$l_i = \arg\max_{y \in Y}[\alpha_i(y)\beta_i(y)],$$

or (2) MAP estimate by using Viterbi algorithm as defined in Equation (2) except for using a maximization operation in place of summation [see Rabiner (1989) for details].

For score combination, we define two kinds of features: one is the score feature

$$f_j^{\text{score}}(x_i, y_i) = \begin{cases} S_j(x_i) & \text{if } y_i = j \\ 0 & \text{otherwise.} \end{cases}$$

and the other is the transition feature

$$f_{<j,k>}^{\text{trans}}(x_i, y_i, y_{i-1}) = \begin{cases} P(y_i|y_{i-1}) & \text{if } y_i = j, y_{i-1} = k \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where $j, k \in Y = \{H, E, C\}$. $P(y_i|y_{i-1})$ can be learned from the training data:

$$P(y|y') = \frac{\#\text{ of occurences } y'y}{\#\text{ of occurences } y'}.$$

We notice that for MEMMs, the transition information is already encoded implicitly in the Viterbi process. Since MEMMs have the advantage of allowing as many features as possible without decreasing the performance, we also treat f^{trans} as explicit features in case they help.

Higher-order Markov models As shown in Figure 3B, MEMMs have first-order Markov assumption, i.e. $P(y_{i+1}|y_i) = P(y_{i+1}|y_i, y_{i-1})$. The effect is 2-fold: on one hand, it simplifies the model and dramatically reduces the computational cost; on the other hand, this assumption is clearly inappropriate for secondary structure prediction, where the structure dependencies extend over several residues and even involve long-distance interactions. To solve this problem, higher-order Markov models (HOMEMMs) can be applied (Rabiner, 1989).

For simplicity, we only consider second-order Markov models, in which the next state depends upon the two previous states (Fig. 4B). The second-order Markov models can be transformed to an equivalent first-order Markov model by redefining the state $\hat{y}_i$ as

$$\hat{y}_i = \langle y_i, y_{i-1} \rangle \in Y \times Y = \Omega.$$

In secondary structure prediction, the set of new states is $\Omega = \{\text{HC, HE, HH, EC, EE, EH, CC, CE, CH}\}$. We notice that the number of states grows exponentially.

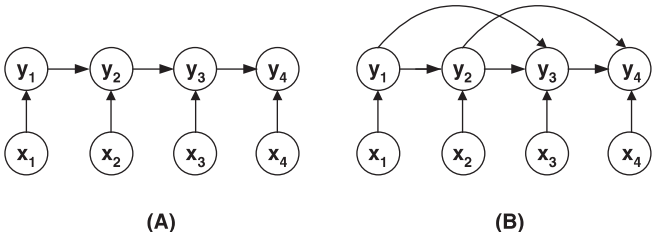

Fig. 4. Graphical structures of (**A**) MEMM and (**B**) second-order MEMM.

The score feature is the same as discussed above and the transition feature is defined as follows:

$$P(\hat{y}_i|\hat{y}_{i-1}) = P(\langle y_i, y_{i-1}\rangle|\langle y_{i-1}, y_{i-2}\rangle) = P(y_i|y_{i-1}, y_{i-2}). \quad (4)$$

Pseudo state duration markov models Higher-order Markov models provide a solution to circumvent the state independence assumptions. However, the number of new states $\|\Omega\|$ is an exponential function of the order K. The computational costs become intractable as k increases. To solve the problem, we devise a heuristic method that is able to encompass more history information with the same computational cost as one-order Markov models, namely pseudo state duration Markov models (PSMEMMs).

Our heuristics are based on the observation that the distribution of the segment length varies for different structures, as shown in Figure 5A (only segments less than 20 residues are shown). From the graph, we can see that different segment lengths are preferred by different structures. For example, around 25% of beta-strands have only one residue, which are in fact beta-bridges; there are also short 3_{10}-helices with three or four residues.

To incorporate such kind of information, we define $P(y|y', N)$ as the probability that the current state is y given the recent history of N consecutive y'. $P(y|y', N)$ is learned from the training data in the following way:

$$P(y|y', N) = \frac{\#\text{ of occurences } \bar{y'}\overbrace{y'y'\cdots y'}^{N}y}{\#\text{ of occurences } \bar{y'}y'y'\cdots y'}.$$

The distribution of $P(H|E, N)$, $P(H|H, N)$ and $P(E|E, N)$, $P(E|H, N)$ for $N \leq 20$ is plotted in Figure 5B and C, respectively (we assume there is no direct transition from H to E, or from E to H). Data sparsity problems might occur when N grows larger. It can be addressed by smoothing methods, such as Laplace smoothing.

All the algorithms and definitions are similar as MEMMs except that the transition feature is:

$$f_{<j,k>}^{\text{trans}}(x_i, y_i, y_{i-1}) = \begin{cases} P(y_i|y_{i-1}, N) & \text{if } y_i = j, y_{i-1} = k \\ 0 & \text{otherwise,} \end{cases}$$

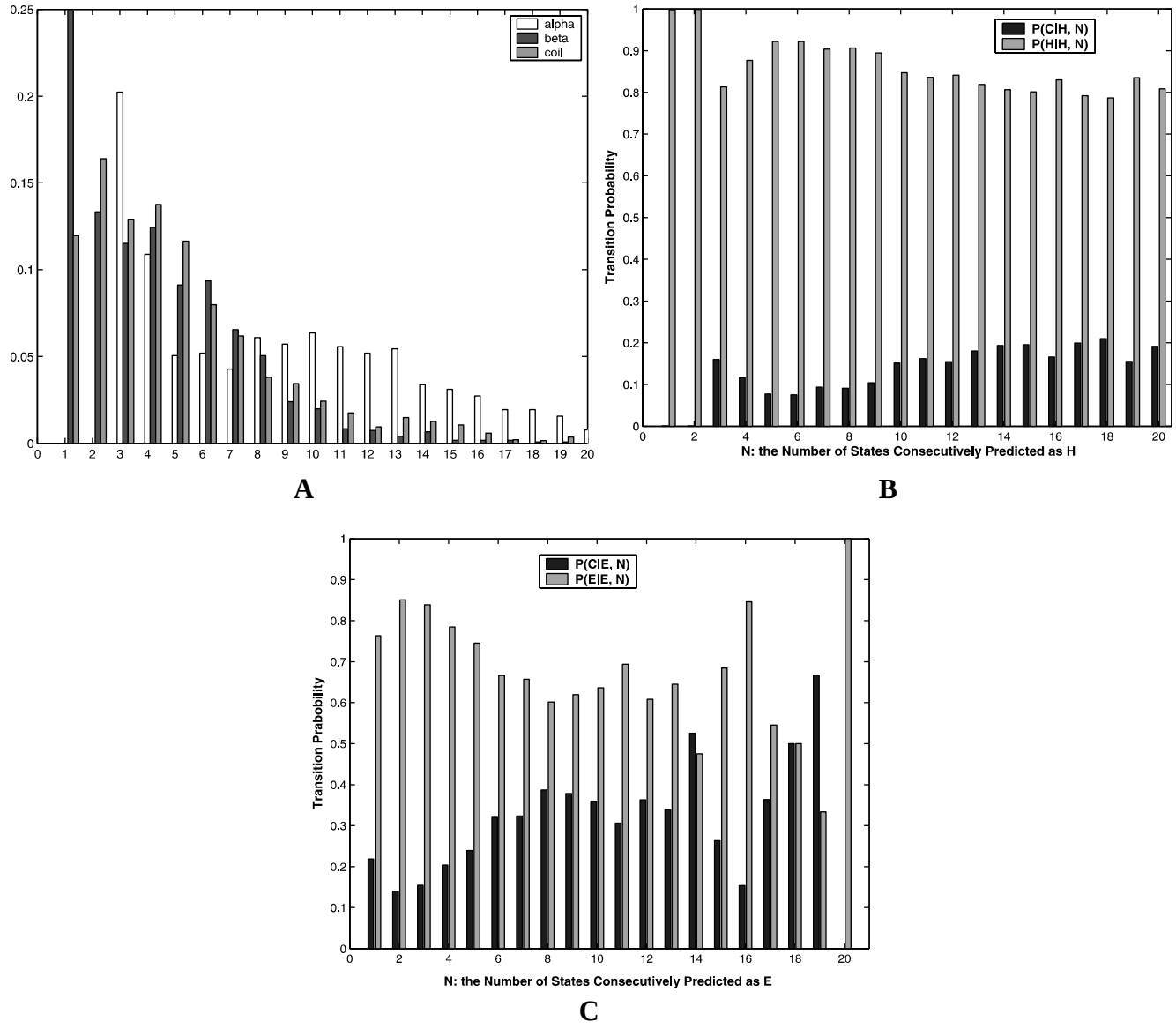

Fig. 5. (**A**) The distribution of the segment length for different structures; (**B**) the transition probability of helices; (**C**) the transition probability of beta-sheet.

where N can be back-traced from the maximization histories in the dynamic programming process.

Conditional random fields In addition to Markov assumption, MEMMs also suffer from the problem known as the *label bias* problem. In short, the *label bias* means that the total probability 'received' by y_{i-1} must be passed on to labels y_i at time i even if x_i is completely incompatible with y_{i-1} [see Lafferty *et al*. (2001) for full discussion]. CRFs proposed by Lafferty *et al*., are a globally normalized extension to MEMMs that avoid the label bias problem (Lafferty *et al*., 2001).

CRFs are *undirected* graphical models (also known as *random fields*) and calculate the conditional likelihood $P(\mathbf{y} \mid \mathbf{x})$ directly. The graphical structure for chain-form CRFs is shown in Figure 3C. By Hammersely–Clifford theorem (Hammersley and Clifford, 1971) and using exponential model, the conditional probability $P(\mathbf{y}|\mathbf{x})$ is defined as

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_0} \exp\left[\sum_{i=1}^{N} \sum_{k} f_k(y_{i-1}, y_i, \mathbf{x})\right]. \qquad (5)$$

Similar to MEMMs, f_k can be arbitrary features and the weight λ_k is learned via maximizing the conditional likelihood of the training data.

Comparing Equation (5) with Equation (1) in MEMM, the only difference between the two is that MEMMs take a local

Table 1. Summary of the graphical models

	1st-order Markov	Label bias	Flexibility of features	Global optimum
HMMs	+	+	−	−
MEMMs	+	−	+	−
HOMEMMs	−	−	+	−
PSMEMMs	−	−	+	−
CRFs	+	+	+	+

normalization Z_0 while CRFs are global. This enables CRFs to have convex optimization function so that the global optimal solutions are guaranteed (Lafferty *et al*., 2001). However, it is not straightforward for CRFs to find the optimum quickly. Very recently, the quasi-Newton methods are shown to be significantly more efficient than other methods (McCallum, 2003).

As in MEMMs, the 'forward value' $\alpha_i(y)$ is defined as the probability of being in state y at time i *given* the observation up to time i and $\beta_i(y)$ is the probability of starting from state y at time i *given* the observation sequence after time i. The recursive step is:

$$\alpha_{i+1}(y) = \sum_{y'} \alpha_i(y') \exp\left[\sum_k \lambda_k f_k(y', y, \mathbf{x}, i+1)\right],$$

$$\beta_i(y') = \sum_{y \in Y} \exp\left[\sum_k \lambda_k f_k(y', y, \mathbf{x}, i+1)\right] \beta_{i+1}(y).$$

The forward–backward and Viterbi algorithms can be derived accordingly. The features for score combination are the same as that defined for MEMMs.

Summary Table 1 summarizes the properties of the graphical models discussed above. We can see that all the models except HMMs have the flexibility of including *any* feature and therefore are good for score combination. However, this only indicates the general power of the models; the effectiveness and computational costs will be further discovered in our experiments.

Materials

In our experiments, we used the CB513 dataset by Cuff and Barton (Cuff and Barton, 1999), which many previous papers reported results on (Hua and Sun, 2001; Kim and Park, 2003; Guo *et al*., 2004). It consists of 513 non-homologous protein chains that have an SD score, i.e. Z score for comparison of the native sequences given by $(V - \overline{x})/\sigma$, of less than five (Cuff and Barton, 1999). The dataset can be downloaded from the website http://barton.ebi.ac.uk/.

We followed the DSSP definition for protein secondary structure assignment (Kabsch and Sander, 1983). The definition is based on hydrogen bonding patterns and geometrical constraints. Based on the discussion by Cuff and Barton (1999), the eight DSSP labels are reduced to a three state model as follows: H and G to Helix (H), E and B to Sheets (E) and all other states to Coil (C).

All the combination methods discussed above can be applied to combine predictions from single or multiple systems. To provide accurate evaluation, we choose to use outputs from a single system to distinguish the improvement from considering correlations of labels and long-distance interactions with the improvement from the overlapping information by different systems.

For protein secondary structure prediction, the state-of-art performance is achieved by window-based methods using the PSI-BLAST profiles (Jones, 1999). In our experiments, we apply a linear transformation L to the PSSM matrix elements according to

$$L(x) = \begin{cases} 0 & \text{if } (x \le -5) \\ L(x) = \frac{1}{2} + \frac{x}{10} & \text{if } (-5 \le x \le 5), \\ L(x) = 1 & \text{otherwise.} \end{cases}$$

This is the same transform used by Kim and Park (2003) in the recent CASP (Critical Assessment of Structure Predictions) competition, which achieved one of the best results for protein secondary structure prediction. The window size is set to 13 by cross-validation.

Various measures are used to evaluate the prediction accuracy, including overall per-residue accuracy (Q_3), Matthew's correlation coefficients per structure type (C_H, C_C, C_E) and segment of overlap (SOV) (Rost *et al*., 1994; Zemla *et al*., 1999) and the per-residue accuracy for each type of secondary structure (Q_H, Q_E, Q_C; $Q_\mathrm{H}^\mathrm{pre}$, $Q_\mathrm{E}^\mathrm{pre}$, $Q_\mathrm{C}^\mathrm{pre}$) [see Hua and Sun (2001) for detailed definition]. Seven-fold cross-validation was used, which is the same setting as in Rost and Sander (1993) and Hua and Sun (2001).

RESULTS

Score distribution

One of the assumptions for why combination methods work is that the score contains more information than a single label. If two scores $S_j(x_i)$ and $S_k(x_i)$ for residue x_i are very close, then combining them with the information from neighbors might help the final prediction adjust to the correct label by overriding the small score difference. From the aspect of information theory, we try to use combination methods for error-correction.

Therefore, we studied the distribution of the differences between the maximum score $M(x_i)$ and the second maximum score $M'(x_i)$ for residue x_i, as shown in Figure 6. From the plot, we can see that the probability that the differences D are close to zero is very high. The cases falling into the green area $[P(|D| \le 0.1)]$ covers around 5% of the total residues, which demonstrates that there is still room for improvement by score combination.

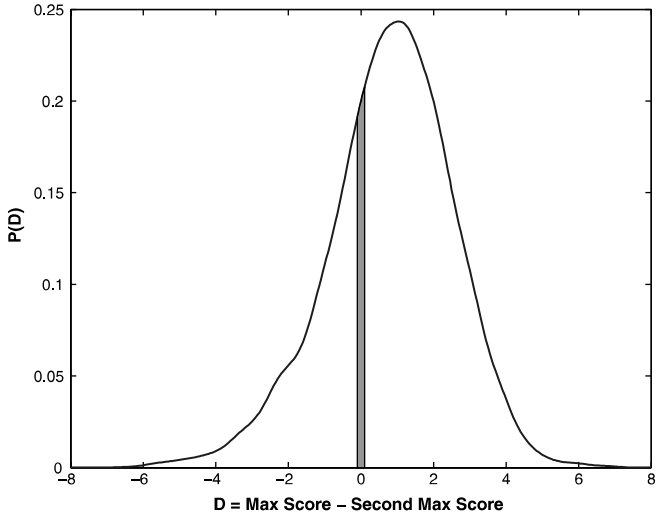

Fig. 6. The distribution of the differences between max score and second max score.

Comparison of combination strategies

To fairly evaluate the effectiveness of different methods, we use the same input, i.e. the score matrix S generated from SVMs with RBF kernels using the PSI-BLAST profiles. For the window-based combination, we use the decision tree algorithm C4.5 (Quinlan, 1993) for label combination and SVMs with RBF kernels for score combination. The window size w is set to 15.

Table 2 lists the results of the window-based methods[1]:

- Generally speaking, the window-based score combination improved the prediction more than the label combination. This confirms our expectation since the scores contain more information than a single label.
- The label combination resulted in maximum improvement for predicting helices rather than other structures. King and Sternberg reported a similar observation and showed that the extracted rules are most relevant to helices (King and Sternberg, 1996).
- The prediction accuracy has increased for both helices and sheets by score combination.

In terms of the graphical models for score combination, we examined the four methods discussed before. To fairly compare with window-based methods, only two kinds of features are used for the prediction: the score features f^{score} and the transition features f^{trans}, although we believe incorporating other features will improve the predictions more. For higher-order MEMMs, we choose the second-order MEMMs as representative. To get the optimum λ_k in MEMMs and CRFs, the conjugate gradient algorithm was applied (the code can be downloaded from http://www.cs.toronto.edu/∼buescher/). Table 3 shows the results of the four graphical models for score combination:

- Generally speaking, the graphical models for score combination are consistently better than the window-based approaches, especially in SOV measure.
- For the MEMMs, the prediction accuracy using Viterbi algorithm is better than that using marginal mode. It is interesting to note that the opposite is true for CRFs.
- Compared with MEMMs, HOMEMMs and PSMEMMs were somewhat improved in SOV measure since these methods consider more history information. However, there is little difference in performance between HOMEMMs and PSMEMMs. This might indicate that higher-order MEMMs will hardly add more value than second-order MEMMs.
- CRFs perform the best among the four graphical models. They exhibit moderate improvements for predicting helices and especially sheets. Global optimization and removing label bias seem to help since these are the only differences between MEMMs and CRFs.

Table 4 summarizes our discussion above and provides a qualitative estimation of computational costs as well as the performance for each method.

Combination bounds using PSI-BLAST profiles

We have discussed several combination strategies using graphical models and our experiments demonstrate that those methods can improve the secondary structure prediction performance to a certain extent. However, what is the best performance we can get by combining the predictions? Answering this question will involve much deeper analysis and more thorough experiments. However, we can get a rough idea of the limits by providing the location of the true segment boundaries.

Two simple strategies have been used: the max rule, i.e. assigning the label j with the maximum score $\max_{i,j} S_j(x_i)$ to all residues within the segment, and the sum rule, i.e. assigning the label j with the maximum sum of scores $\max_j[\sum_i S_j(x_i)]$ to all the residues within the segment (Table 5). Since no method can predict the segment with perfect accuracy, these results can be seen as an upper bound by using PSI-BLAST profiles. From the results, we can see that even given the true segment assignments, we are still far from reaching an accuracy of 90% using current PSI-BLAST profile features. The ideal solution would be to incorporate other informative non-local features, by which the graphical models can gain more improvements.

[1]The results for window-based score combination using SVMs are slightly better than the results reported in Guo *et al*. (2004) on the same dataset.

Table 2. Results of protein secondary structure prediction on CB513 dataset using window-based combination methods

Combination method	SOV (%)	Q_3 (%)	Q_H (%)	Q_C (%)	Q_E (%)	Q_H^{pre} (%)	Q_C^{pre} (%)	Q_E^{pre} (%)	C_H	C_C	C_E
None	75.6	76.7	78.0	83.2	62.7	83.6	72.1	77.2	0.71	0.58	0.62
Dtree	75.7	76.7	78.0	83.2	62.8	**83.7**	72.1	77.1	**0.72**	0.58	0.62
SVM	75.7	76.9	**81.4**	76.7	**70.5**	82.1	75.2	72.2	**0.72**	0.58	**0.63**

Table 3. Results on CB513 dataset using different combination strategies. MEMM^p, CRF^p: p refers to different way to compute the labels; $p = 1$: marginal model; $p = 2$: Viterbi algorithm

Combination method	SOV (%)	Q_3 (%)	Q_H (%)	Q_C (%)	Q_E (%)	Q_H^{pre} (%)	Q_C^{pre} (%)	Q_E^{pre} (%)	C_H	C_C	C_E
None	75.6	76.7	78.0	83.2	62.7	83.6	72.1	77.2	0.71	0.58	0.62
MEMM[1]	75.6	76.7	77.8	83.6	62.1	83.7	71.8	77.8	0.71	0.58	0.62
MEMM[2]	**76.0**	76.8	78.2	83.4	62.2	83.7	72.0	**78.0**	0.71	0.58	0.62
HOMEMMs[2]	**76.1**	**76.9**	**78.3**	83.4	62.4	83.6	72.1	77.9	0.71	0.59	0.62
PSMMEMMs[2]	**76.1**	**76.9**	**78.3**	83.3	62.2	83.6	72.0	**78.0**	0.71	0.58	0.62
CRF[1]	**76.2**	**77.0**	**78.3**	83.4	**63.4**	83.7	72.1	**78.0**	**0.72**	0.58	**0.63**

Table 4. Summary of computational costs and effectiveness for different combination strategies. H/L/M: high/low/medium computational costs; +/−: improvement/no improvement over the baseline results without combination

	Train	Test	Helices	Sheets	Coil	Segment
DTree	*M*	*L*	+	−	−	−
SVM	*H*	*H*	+	+	−	−
MEMMs	*H*	*L*	−	−	−	+
HOMEMMs	*H*	*L*	−	−	−	+
PSMEMMs	*H*	*L*	−	−	−	+
CRFs	*H*	*L*	+	+	−	+

Table 5. Results of combination given the location of each structure segment on CB513 dataset by seven-fold cross-validation

Combination method	Q_3 (%)	Q_E (%)	Q_E^{pre} (%)	C_E
Baseline	76.7	62.7	77.2	0.62
Sum rule	85.9	73.5	91.1	0.77
Max rule	83.2	69.0	89.4	0.73

CONCLUSIONS

In this article, we analyzed current secondary structure prediction methods and identified the combination problem for sequences: how to combine the predicted scores or labels from a single or multiple systems with the consideration of neighbors and long-distance interactions. We studied previous work that uses window-based combination methods and proposed to use powerful graphical chain models to improve the combination. Our experiments show that graphical models are consistently better than the window-based methods. In particular, CRFs improve the predictions for both helices and sheets, while sheets benefitted the most.

Our goal is to evaluate different combination methods and provide a deeper understanding of how to effectively improve secondary structure prediction. Although our discussion is focused on combining predictions from a single secondary structure prediction system, all the methods discussed can be applied to combine results from different systems and include other physico-chemical features. Since each part in a secondary structure prediction system is not independent (Fig. 1), our future work would be to consider all parts as a whole and build a hybrid system.

ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation under Grant No. 0225656.

REFERENCES

Bystroff,C., Thorsson,V. and Baker,D. (2000) HMMSTR: a hidden Markov model for local sequence-structure correlations in proteins. *J. Mol. Biol.*, **301**, 173–190.

Crooks,G.E. and Brenner,S.E. (2004) Protein secondary structure: entropy, correlations and prediction. *Bioinformatics*, **20**, 1603–1611.

Cuff,J.A. and Barton,G.J. (1999) Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins*, **34**, 508–519.

Cuff,J.A. and Barton,G.J. (2000) Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins*, **40**, 502–511.

Guermeur,Y., Geourjon,C., Gallinari,P. and Deleage,D. (1999) Improved performance in protein secondary structure prediction by inhomogeneous score combination. *Bioinformatics*, **15**, 413–421.

Guo,J., Chen,H., Sun,Z. and Lin,Y. (2004) A novel method for protein secondary structure prediction using dual-layer SVM and profiles. *Proteins*, **4**, 738–743.

Hammersley,J. and Clifford,P. (1971) Markov fields on finite graphs and lattices. Unpublished manuscript.

Hua,S. and Sun,Z. (2001) A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *J. Mol. Biol.*, **308**, 397–407.

Jones,D.T. (1999) Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.*, **292**, 195–202.

Kabsch,W. and Sander,C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **22**, 2577–2637.

Karplus,K., Barrett,C. and Hughey,R. (1998) Hidden Markov models for detecting remote protein homologies. *Bioinformatics*, **14**, 846–856.

Kim,H. and Park,H. (2003) Protein secondary structure prediction based on an improved support vector machines approach. *Protein Eng.*, **16**, 553–560.

King,R.D. and Sternberg,M.J. (1996) Identification and application of the concepts important for accurate and reliable protein secondary structure prediction. *Protein Sci.*, **5**, 2298–2310.

Krogh,A. and Sollich,P. (1997) Statistical mechanics of ensemble learning. *Phys. Rev. E.*, **55**, 811–825.

Lafferty,J., McCallum,A. and Pereira,F. (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. *Proceedings of the 18th International Conference on Machine Learning (ICML'01)*, 282–289, Williamstown, MA, USA.

McCallum,A., Freitag,D. and Pereira,F. (2000) Maximum entropy Markov models for information extraction and segmentation. *Proceedings of the 17th International Conference on Machine Learning (ICML'00)*, 591–598, Stanford, CA, USA.

McCallum,A. (2003) Efficiently inducing features of conditional random fields. *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI'03)*, 403–410, Acapulco, Mexico.

Ouali,M. and King,R.D. (2000) Cascaded multiple classifiers for secondary structure prediction. *Protein Sci.*, **9**, 1162–1176.

Pollastri,G., Przybylski,D., Rost,B. and Baldi,P. (2002) Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins*, **47**, 228–235.

Quinlan,J.R. (1993) *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Francisco, CA, USA.

Rabiner,L.R. (1989) A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, **77**, 257–286.

Rost,B. and Sander,C. (1993) Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.*, **232**, 584–599.

Rost,B., Sander,C. and Schneider,R. (1994) Redefining the goals of protein secondary structure prediction. *J. Mol. Biol.*, **235**, 13–26.

Rost,B. (2001) Review: protein secondary structure prediction continues to rise. *J. Struct. Biol.*, **134**, 204–218.

Salamov,A.A. and Solovyev,V.V. (1995) Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments. *J. Mol. Biol.*, **247**, 11–15.

Selbig,J., Mevissen,T. and Lengauer,T. (1999) Decision tree-based formation of consensus protein secondary structure prediction. *Bioinformatics*, **15**, 1039–1046.

Sollich,P. and Krogh,A. (1996) Learning with ensembles: how over-fitting can be useful. *Proceedings of Advances in Neural Information Processing Systems, (NIPS'96)*, 190–196, Denver, CO, USA.

Vapnik,V.N. (1995) *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, NY, USA.

Wolpert,D.H. (1992) Stacked generalization. *Neural Networks*, **5**, 241–259.

Zemla,A., Venclovas,C., Fidelis,K. and Rost,B. (1999) A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment. *Proteins*, **34**, 220–223.