Evolutionary Bioinformatics

Libertas Academica
FREEDOM TO RESEARCH

ORIGINAL RESEARCH

Profiles and Majority Voting-Based Ensemble Method for Protein Secondary Structure Prediction

Hafida Bouziane, Belhadri Messabih and Abdallah Chouarfia

Department of Computer Science, USTO-MB University, BP 1505 El Mnaouer, Oran, Algeria.
Corresponding author email: h_bouziane@univ-usto.dz; messabih@univ-usto.dz; chouarfia@univ-usto.dz

Abstract: Machine learning techniques have been widely applied to solve the problem of predicting protein secondary structure from the amino acid sequence. They have gained substantial success in this research area. Many methods have been used including k-Nearest Neighbors (k-NNs), Hidden Markov Models (HMMs), Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs), which have attracted attention recently. Today, the main goal remains to improve the prediction quality of the secondary structure elements. The prediction accuracy has been continuously improved over the years, especially by using hybrid or ensemble methods and incorporating evolutionary information in the form of profiles extracted from alignments of multiple homologous sequences. In this paper, we investigate how best to combine k-NNs, ANNs and Multi-class SVMs (M-SVMs) to improve secondary structure prediction of globular proteins. An ensemble method which combines the outputs of two feed-forward ANNs, k-NN and three M-SVM classifiers has been applied. Ensemble members are combined using two variants of majority voting rule. An heuristic based filter has also been applied to refine the prediction. To investigate how much improvement the general ensemble method can give rather than the individual classifiers that make up the ensemble, we have experimented with the proposed system on the two widely used benchmark datasets RS126 and CB513 using cross-validation tests by including PSI-BLAST position-specific scoring matrix (PSSM) profiles as inputs. The experimental results reveal that the proposed system yields significant performance gains when compared with the best individual classifier.

Keywords: protein secondary structure prediction, k-Nearest Neighbors, feed-forward Neural Networks, Multi-class Support Vector Machines (M-SVMs), ensemble method, Position-Specific Scoring Matrix (PSSM) profiles

Evolutionary Bioinformatics 2011:7 171–189

doi: 10.4137/EBO.S7931

This article is available from http://www.la-press.com.

Introduction

Gaining information about proteins both structurally and functionally remains an ultimate goal in biological and medical research due to their importance in living organisms of which they are the major components. Proteins are large and complex organic compounds that consist of amino acids joined by peptide backbones. The poly-peptide chains thus constituted are called primary structures that can fold into complicated three-dimensional (3-D) structures (native structures) which largely describe their functions. Thus, it is essential to know the protein's three-dimensional structure so as to infer its function. With recent advances in large genome sequencing projects, there is an increasing need to determine this structure. The number of protein sequences deposited in the PDB[a] continues to grow much faster than the number of known protein structures and the major interest in current time is then to bridge this ever-widening gap between sequences and structures. Amino acid sequence (primary or 1-D structure) contains sufficient information specifying the three-dimensional structure.[1] However, structure determination from known sequence is not a straightforward task. It is laborious, expensive, very time-consuming and sometimes, even impossible to use purely experimental techniques such as X-ray crystallography and Nuclear Magnetic Resonance spectroscopy. Thus, in silico methods present alternative approaches to accomplish this task with low cost and reduced time and effort. Therefore many methods have been rigourously explored to perform the essential intermediate step on the way to predicting this structure, which is to predict the secondary (2-D) structure from the primary structure. The secondary structure (SS) consists of local folding regularities maintained by hydrogen bonds. Three classes (patterns) characterize the SS: α-helices, β-sheets (Extended-strands) and coils (the remaining non regular conformations). Given a protein sequence, the Protein Secondary Structure Prediction (PSSP) problem is to predict whether each amino acid also known as residue is in either α-helix (H), β-sheet (E) or coil (C) state. From the point of view of pattern recognition, this can be seen as a 3-class discrimination problem.

Since the 1970s, many approaches for PSSP have been developed. The first efforts have been made for predicting the SS solely from the amino acid sequence using simple linear statistics and expert rules taking into account only the physico-chemical properties of single amino acids.[2–4] The average tree-state per-residue score Q_3 (a prediction accuracy measure that gives the percentage of correctly predicted secondary structures) of these methods referred to as the first generation methods was in the range 50%–54%. Around 1988, the PSSP methods have been extended in various ways to include correlations among amino acids.[5,6] The first use of ANNs by Qian and Sejnowski[7] improved the Q_3 score to 62%. This second generation of sequence alone prediction methods have not been sufficient to achieve successful PSSP, and it was claimed that predictions cannot be better than 65(±2)%.[8] The PSSP methods have improved substantially since 1990 through the use of the evolutionary information exploiting the information coming from homologous sequences. The use of multiple alignments of protein sequences instead of single amino acid sequences revolutionised SS prediction. With this third generation of PSSP methods, the Q_3 score exceeded 70%. Advances in computing techniques and the availability of large families of homologous sequences have led to methods that are generally available via the web. The flagship for this generation of methods was PHD[8,9] which has been inspired by the basic architecture of Qian and Sejnowski. It improved the prediction accuracy to over 70%. Other profiles-based methods also became available such as SOPMA,[10] DSC,[11] NNSSP[12] and PREDATOR.[13] Recently, there have been approaches which achieve even higher accuracy (>75%) using consensus of the existing methods to prediction refinement. The methods have gradually improved in accuracy in the works of Riis and Krogh,[14] Jones[15] with PSIPRED, Cuff and Barton[16] with Jpred, Baldi et al[17] with SSpro, Pollastri et al with Porter[18] an evolution of SSpro, Bondugula and Xu[19] with MUPRED, Karplus et al[20] with SAM-T99sec and Petersen et al[21] with GOR V. The predictive quality of these availible PSSP servers is evaluated and compared within the frameworks of several initiatives including CASP[b] (Critical Assessment Structure Prediction), CAFASP[c] (Critical Assessment of Fully Automated Structure Prediction) and EVA[d] (EValuation of Automatic protein structure prediction). PSSP

[a] Brookhaven Protein Data Bank of solved structures availible at http://www.rcsb.org/pdb/.

[b] http://predictioncenter.org/.

[c] http://www.cs.bgu.ac.il/dfischer/CAFASP5/.

[d] http://cubic.bioc.columbia.edu/eva/.

methods based on Support Vector Machines have also been developed and have been demonstrating good performance.[22–28] An overview of the earliest SVM-based PSSP methods can be found in.[29]

Currently the performance of all the best PSSP methods in term of Q_3 score is between 75% and 80% depending on the training and the test datasets. All the recent above-mentioned methods use homology as the important factor to determine the SS and consensus to improve the prediction quality. Thus it would make sense to tackle the PSSP problem using these two factors. The concept of combining classifiers to benefit from their strengths so as to improve the performance has long been established. However, the choice of an effective combiner may be a difficult task in addition of the problem of generating a set of classifiers with reasonably good individual performance and independent (no biased) predictions.

In this study, we investigate how performance can be enhanced by combining k-Nearest Neighbors, Artificial Neural Networks and Multi-class Support Vector Machines, incorporating position-specific scoring matrix (PSSM) profiles to predict the SS of globular proteins. To do so, two variants of the majority voting rule are used. Simple Majority Voting (SMV) which counts the votes and allocates a queried residue to the class that gains the majority votes and Weighted Majority Voting (WMV) which weights each vote by the corresponding classifier prediction accuracy. If two or more classes gain the same vote (conflicting decision), SMV uses two strategies. The first one consists in the traditional scheme of SMV which chooses one of the classes arbitrarily (SMV) and the second scheme that we used and named in our experiments "Influenced Majority Vote" (IMV), assigns the class predicted by the best classifier in the ensemble. We also used a heuristic based filter to refine the predictions obtained by each scheme of the ensemble method proposed.

The remainder of this paper is organized into three sections starting with section 2 which describes in detail the general framework of the proposed system. The experimental results are summarized in section 3, followed by a conclusion and an overview of future work drawn in section 4.

Materials and Methods

This section describes the general framework used for achieving the goal of the analysis described in this article. It first introduces the PSSP problem, presents the datasets used and then explains the detail of methods and algorithms used to implement and test the proposed system in predicting the secondary structures of globular proteins.

Data encoding and PSSP problem formulation

Within known protein structures, the most frequent secondary structures are α-helices and β-sheets. Beside these two common structures, six other rare types of structures are further proposed by Dictionary of Secondary Structures of Proteins program (DSSP)[30] which are explicitly *I* (π-*helix*), G (3_{10}-helix), B (*isolated* β-bridge), T (hydrogen bonded turn), S (bend) and the rest, which leads to a total of eight types. These eight types of structures can be grouped into three larger classes: α-helices (H), β-sheets (E), and coils (C) using assignment scheme. Standard assignments do not exist, so defining the bondaries between helix, strand and coil structures is arbitrary. The assignment method influences the prediction accuracy,[16] so one generally tends to use a translation scheme which leads to higher estimates of accuracy. In addition to the DSSP method which is adopted by PHD,[8] other reduction methods have been proposed in the literature such as STRIDE,[31] DEFINE,[32] KAKSI[33] and so on. In this study, we concentrate exclusively on the DSSP method since it has been the most widely used reduction method. One can find five main DSSP assignment schemes which are explicitly: (1) H,G and I to H; E to E; all other states to C; (2) H,G to H; E,B to E; all other states to C; (3) H,G to H; E to E; all other states to C; (4) H to H; E,B to E; all other states to C; (5) H to H; E to E; all other states to C. Here, we adopted the scheme (2) since it is usually used in many studies. This assignment scheme treats B (*isolated* β-*bridge*) as part of a β-sheet (E) which increases the proportion of state (E).

The primary structure which is described as a sequence of amino acids in the polypeptide chain can be represented as a string on the finite alphabet A with $|A| = 20$ (number of naturally occurring amino acids).

Let $A = \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, Y, V, W\}$ be a set of all amino acids, each letter corresponding to a different amino acid. The prediction of the secondary structure can be described as the following mapping:

$$\{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S,\\ T, Y, V, W\}^n \overset{PSSP}{\rightarrow} \{H, E, C\}^n \tag{1}$$

Each amino acid in the sequence of length n belongs to one of the three structural conformations H, E or C. A large number of methods for ensuring such mapping have been developed and nowadays the perfect discrimination of the three classes (H, E, C) remains a challenge. A computational method which can effectively translate the sequence into an accurate structure is urgently required. The major methods based on supervised learning have adopted a windowing technique for generating training and testing sequences. The window consisting of fixed number of amino acids centred on the residue to be predicted (queried residue) is used to incorporate the influence of the neighbors into the prediction. To generate different input sequences, the window slides over the protein sequence, and one amino acid is considered at a time. The output value represents the secondary structure predicted of the queried residue. Many studies have demonstrated that the window size influences the quality of the prediction. Qian and Sejnowski[7] varied the window size from 1 to 21 and empirically found 13 to be the most effective. In this study, we used the same window size.

PSSM generation

Multiple alignments can produce position-specific profiles, which provide crucial information about structure. Using position-specific profiles as inputs for structure prediction can improve 5%–10% the prediction accuracy than the sequence alone.[8] Position-specific profiles are mainly generated by position-specific iterated BLAST (PSI-BLAST) searches[34] and Hidden Markov Models.[35] The multiple alignment is converted into a profile: for each position, the vector of amino acid frequencies is calculated based on the alignment. Here, we performed PSI-BLAST to obtain the profile for each sequence, setting the parameter *j* (number of iterations) to 3, using an *e* value of 0.01 with the NCBI's nr[e] database as the sequence database. The PSSM has $20 \times n$ elements, where n is the length of the target sequence and each element represents the log-likelihood of particular residue substitution based on a weighted average of BLOSUM62[36] matrix scores for a given alignment position in the template. The profile matrix elements obtained in the range ±7 are scaled to the required 0–1 range by using LIBSVM software.[37] The profiles obtained were then used as inputs to each classifier. Prediction at a given position depends on amino acid frequencies in the profile at the position and neighboring positions within a range defined by the window of size 13 in our experiments, then each input vector has 20 × 13 components.

Training and test datasets

Many datasets have been used in the PSSP experiments and different prediction accuracies have been reported. Predicting the structure of small proteins is naturally easier and faster than predicting the structure of large proteins. In this study, we experiment on and test the proposed system on sequences of different sizes using two widely used datasets. The first one is the dataset of 126 globular protein chains proposed by Rost and Sander,[8] referred to as the RS126 dataset. It has been extracted from the initial RS130 dataset (excluding the four membrane protein chains 1*pre_C*, 1*pre_H*, 1*pre_L*, 1*pre_M*). It is a non-homologous dataset, that no two proteins in the set share more than 25% sequence identity over a length of more than 80 residues. It contains 23349 residues with 32% α-helix, 23% β-sheet, and 45% coil. The second dataset has been constructed by Cuff and Barton[16] referred to as CB513 dataset. It has 513 non-homologous protein chains with a total of 84119 residues and has the following distribution: 36.4% α-helix, 22.91% *β*-sheet and 41.5% coil. It includes the CB396 dataset and almost all sequences of RS126 dataset except eleven protein chains that have an SD score (pairwise sequence similarity measure) of at least five.[16,15] In order to estimate the generalization error, methods are typically tested using k-fold cross-validation, where a dataset is split into k subsets. In each step of the cross-validation, k − 1 of them are used for training and the remaining one for testing. The process is repeated k times, until all the k subsets are used once for testing. The prediction accuracy is estimated by calculating the average accuracy across all the k steps. In this study, we used the seven-fold cross-validation on both RS126 and CB513 datasets.

[e] ftp://ftp.ncbi.nih.gov/blast/db.

Ensemble method

In this section, we first describe the proposed ensemble method, starting with the ensemble members and then we introduce the three voting combination schemes used in our experiments SMV, IMV and WMV. The heuristic based-filter used to refine the predictions is also described in this section.

K-Nearest Neighbor classifier

The k-Nearest Neighbor (k-NN) algorithm classifies an example by assigning it the label most frequently represented among the *k* nearest examples which are closest examples according to a distance-based weighting (Euclidean, Manhattan, etc). The example is classified by a majority vote of its neighbors. The strategy is that classes with the more frequent examples tend to dominate the prediction of the new example. Proper choice of the parameter k depends on the data. In practice, k is usually chosen to be odd. Many studies attempt to find a variant of k-NN rule and appropriate distance measure that improve the performance of the k-NN algorithm. In our study, instead of using the Euclidean distance to measure the distance between examples, we used the algorithm CPW (Class and Prototype Weights learning) proposed in[38] with default parameters. The algorithm uses a distance weighting scheme that will lead to better prediction accuracy than the traditional k-NN classifier. CPW learns the corresponding weights by gradient descent algorithm based on update equations which are explicitly derived by (approximately) minimizing the leaving-one out classification error of the training set.

Feed-forward Neural Networks

The Multi-Layer Perceptron (MLP) and the Radial Basis Function Neural Network (RBFNN) are the most commonly used feed forward Neural Network models in computational biology. In PSSP, they have produced the most accurate SS for the majority of the past few years. The MLP is an improvement of the Perceptron[39] including one or more transition layers known as hidden layers. The units in the input layer are connected to units in the hidden layers, which in turn are connected to units in the output layer. Each connection is associated with a weight. The MLP units take a number of real-valued inputs and generate a single real-valued output, according to an activation function (transfer function) applied to the weighted sum of the outputs of the units in the preceding layer. The most commonly used activation function in this network is a sigmoid function.[40] The learning algorithm can be expressed using generalized Delta rule and back propagation gradient descent.[41] In the RBFNN each hidden unit implements a radial activated function, whose outputs are inversely proportional to the distance from the center of the units. In pattern classification applications the most used radial activated function is the Gaussian.[42,43] The Gaussian's centers influence critically the performance of the RBFNN. Poggio and Girosi[43] showed that using all the training data as centers may lead to network overfitting as the number of data becomes too large, and the gradient descent approach used to update the RBFNN centers moved the centers towards the majority of the data. To avoid these situations, they suggested the use of a clustering algorithm to position the centers.

For both the two NNs, the global error E at the output layer can be either a sum of squared differences of the desired outputs d_i and the actually calculated outputs o_i of each output unit i, and can be expressed as:

$$E = \sum_i (d_i - o_i)^2 \tag{2}$$

Or a log-likelihood cost function defined as:

$$E = \sum_i d_i log\left(\frac{d_i}{o_i}\right) \tag{3}$$

The MLP in this study consists of an input layer, a hidden layer and an output layer. It is trained using the standard back-propagation algorithm. The hidden and the output layer units have sigmoid activation function. For the RBFNN, we have adopted QuickRBF[f] package[44] which uses an efficient least mean square error method to determine the weights associated with the links between the hidden layer and the output layer based on the Cholesky decomposition method. It is capable of delivering the same level of prediction accuracy as the SVM classifier.

Support vector machines and multi-class support vector machines

Recently Support Vector Machines (SVM) technique has been applied successfuly in many applications. Up to now the highest accuracy is achieved by approaches using it. Designed by Vladimir Vapnik

[f] http://muse.csie.ntu.edu.tw/yien/quickrbf/index.php.

and his colleagues[45,46] as a direct implementation of the Structural Risk Minimisation (SRM) inductive principle.[47] It was originally designed for binary (two-class labels) classification. SVMs look for the optimal separating hyperplane between two classes by maximizing the margin between the classes. Basically, SVMs can only solve binary classification problems, they treat the linear case (optimal hyperplane) and the non-linear case by introducing of kernel satisfying Mercer's conditions.[48] In the non-linear case, non-negative slack variables ξ_i have been introduced to characterize the empirical risk (classification error). Let us consider $x \in \mathbb{R}^N$ a point in a hyperspace of dimension N, so a class of hyperplanes can be defined as $w \cdot x + b = 0$, where $w \in \mathbb{R}^N$ represents the weight vector normal to the hyperplane and $b \in \mathbb{R}$ the distance of the hyperplane to the origin (bias). The SVM solves the following problem:

$$min\frac{1}{2}w^t\cdot w+C\sum_{i=1}^{N}\xi_i \qquad (4)$$

$$\text{s.t. } y_i\left[w^t\cdot x_i+b\right]\geq 1-\xi_i,\ \xi_i\geq 0,\ i=1,\ldots,N. \qquad (5)$$

where x_i is a N-dimensional input (an example of $\mathbb{R}^N$), $y_i \in \{-1, 1\}$a label in two class problem, C a predefined positive real number commonly named soft margin constant which represents the penalty parameter and ξ_i are slack variables.[49] The first term of the objective function given in (4) ensures the regularization by minimising the l_2 norm of the weight coefficients. The second term represents the empirical risk, $\xi_i = 0$ indicates that the examples are perfecty linearly separable whereas a nonzero ξ_i means that the classifier introduced some error on the corresponding example. The constraint given in (5) tries to put each example on the correct side of the hyperplane. In a multi-class problem, an example x can belong to one of $Q \geq 3$ classes and the class label is given as $y \in \{1, \ldots, Q\}$. There are two basic approaches in the literature to solve multi-class problem. The first and commonly used approach consists of decomposing the problem to several independent bi-class problems and combines their results to determine the class label (decomposition methods) such as one vs one[50] (experimented in this study), one vs rest,[51] Directed Acyclic Graph (DAGSVM),[52] and Error Corrected Output Coding (ECOC).[53] The second approach solves directly the problem by extending the standard formulation of the SVM to multi-class case by solving a single optimisation problem using standard Quadratic Programming (QP) optimisation techniques. To briefly describe the Multi-class SVMs (M-SVMs) used in this study, let us introduce a primal formulation of a multi-class problem. We consider then the case of Q-classes classification problems with $Q \geq 3$. Each object is represented by its description $x \in \chi$ (input space). Let us represent by Y the set of the classes y which is identified with the set of indices of the classes: $\{1, \ldots, Q\}$. The link between a description x and a class y can be expressed by a joint feature map denoted by φ: $\chi \rightarrow H_{(\kappa,(\cdot))}$ which ensures a map into the high dimensional feature space H induced by κ, a positive semidefinite function on χ^2 and a canonical dot product $(\cdot)$, where data mapped in this space can be compared by using the similarity mesure κ such as:

$$\forall(x,x')\in \chi^2,\kappa(x,x')=\phi(x)\cdot\phi(x') \qquad (6)$$

The function κ is called kernel. The output function for a class $y \in Y$ can then be defined as:

$$\forall x\in \chi,\forall y\in Y,h_y(x:w,b)=(w,\phi(x,y))+b_y. \qquad (7)$$

with $b=(b_j)_{1\leq j\leq Q}$ and $w=(w_j)_{1\leq j\leq Q}$. Thus for each class, it is associated an hyperplane and the classification function consists of assigning to an examlpe x a class y^* as follows:

$$y^*=\underset{y\in Y}{\text{argmax}}\ h_y(x:w,b) \qquad (8)$$

In the SVM formulation (bi-class case), the training algorithm amounts to finding the optimal values of the couples (w; b) for a given choice of the kernel κ and a soft margin constant C. If these two hyperparameters are properly selected, they produce excellent classification result and good generalization. The two commonly used families of kernels are polynomial kernels and radial basis functions. Polynomial kernels are of the form: $\kappa(x, x') = (x \otimes x' + 1)^p$ where the case $p = 1$ gives a linear kernel. The most common form of radial basis function (RBF) is a Gaussian distribution calculated as: $\kappa(x,x')=e^{-\gamma\|x-x'\|^2}$ with $\gamma > 0$. Beyond these two families, there has been interesting work developing other kernels. Let us now describe the

M-SVMs of our interest. The two M-SVMs are the one of Weston and Watkins (M-SVM$_{WW}$)[54] and the one of Crammer and Singer (M-SVM$_{CS}$).[55] The machines share the same architecture but they exhibit distinct properties. A simple introduction of the two M-SVMs is presented here by giving only the primal formulation of the training algorithm for each M-SVM, further details for each machine can be found in the references below.

Problem 1: Weston and Watkins M-SVM (M-SVM$_{WW}$)[54]

$$\min_{h\in H}\left\{\frac{1}{2}\sum_{k=1}^{Q}\| w_k \|^2 + C\sum_{i=1}^{N}\sum_{k\neq y_i}\xi_{ik}\right\} \tag{9}$$

$$\text{s.t.}\begin{cases}\left\langle w_{yi}-w_k,\Phi(x_i)\right\rangle + b_{yi}-b_k \geq 1-\xi_{ik}, & (1\leq i\leq N), (1\leq k\neq y_i\leq Q)\\ \xi_{ik}\geq 0, & (1\leq i\leq N), (1\leq k\neq y_i\leq Q)\end{cases}$$

Problem 2: Crammer and Singer M-SVM (M-SVM$_{\mathrm{CS}}$)[55]

$$\min_{h\in H}\left\{\frac{1}{2}\sum_{k=1}^{Q}\| wk \|^2 + C\sum_{i=1}^{N}\xi_i\right\} \tag{10}$$

$$\text{s.t.}\left\langle (w_{yi}-w_k,\Phi(x_i))\right\rangle + \delta_{yi,k} \geq 1-\xi_i,$$
$$(1\leq i\leq N),\quad (1\leq k\leq Q),$$

where δ is the Kronecker symbol.

LIBSVM

LIBSVM software[37] supports various SVM formulations for classification, regression, and distribution estimation. In this study C-Support Vector Classification (C-SVC)[45,46] using one-against-one (pairwise coupling) technique which constructs one SVM for each pair of classes has been used. For a problem with Q classes, $Q(Q-1)/2$ SVMs are trained and a maximum voting strategy selects the correct class. Here three SVMs are constructed (H/E, E/C, and C/H).

Majority voting combination rule

Various studies have used the majority vote for classifier combination. In this study two variants of majority vote have been experimented. Simple Majority Voting (SMV) and Weighted Majority Voting (WMV) will be introduced in detail in this section.

- Simple Majority Voting (SMV)

Let us consider χ a set of N examples and C a set of Q classes. Let us define an algorithm set $S = \{A_{1,} A_{2,} A_M\}$ which contains the M classifiers used for the voting. Each example $x \in \chi$ is assigned to have one of the Q classes. Each classifier will have its prediction for each example. The final class assigned to each example is the class predicted by the majority of classifiers (gaining the majority votes) for this example. This can be formulated as follows. Let $c_l \in C$ denotes the class of an example x predicted by a classifier A_l, and let a counting function F_k defined as:

$$F_k(c_l)=\begin{cases}1 & c_l = c_k\\ 0 & c_l \neq c_k\end{cases} \tag{11}$$

where c_l and c_k are the classes of C. The count of total votes for class c_k can then be defined as:

$$T_k = \sum_{l=1}^{M} F_k(c_l) \tag{12}$$

The predicted class c for an example x using the algorithm set S is defined to be a class that gains the majority vote as:

$$c = S(x) = \underset{k\in\{1,\ldots Q\}}{\operatorname{argmax}}\ T_k \tag{13}$$

If two or more classes gain the same vote (conflicting decision), two strategies are used. The first strategy chooses one of the classes arbitrarily (SMV). The second strategy that we named in our experiments "Influenced Majority Vote" (IMV) chooses the class predicted by the best classifier in the ensemble.

- Weighted Majority Voting (WMV)

Certain classifiers can be more qualified than others. Weighting the decisions of the classifiers by their prediction accuracy can reinforce the decision of those qualified classifiers, what makes it possible to give more importance to their decision in the vote and consequently may further improve the overall performance than that can be obtained by SMV (where all classifiers have identical weights). In WMV, each vote is weighted by the prediction accuracy value of the classifier that we denote here Acc. The count of total votes for a class c_k in (12) can then be redefined as:

$$T_k = \sum_{l=1}^{M} Acc(A_l) \times F_k(c_l) \qquad (14)$$

The class receiving the greatest total weight is chosen.

The three voting schemes are evaluated in this study using different prediction accuracy measures.

Refining the predictions

In addition to combining classifiers, we apply filter based on explicit rules to "clean up" the predictions obtained by the ensemble method. The filter makes the predictions more realistic by excluding such physically unlikely structures. To assess the prediction accuracy, many PSSP methods filtered/smoothed the predictions by using different strategies.[8,12,56] The main condition for obtaining a coherent structure is that the shortest length of the consecutive state H must be three and two for the consecutive state E. Filtering out all isolated helix and strands residues can improve the performance of the method. We have conceived an heuristic based-code to filter the prediction obtained by the ensemble method. We used the filter proposed by Salamov and Solovyev[12] and added the following rules: [EHE] → [EEE], [HEH] → [HHH], [HCH] → [HHH], [ECE] → [EEE], [HEEH] → [HHHH] and all helices of length one or two are converted to coils, all strands of length one are converted to coils.

Prediction accuracy assessment

In this article, the experimental results are reported with several methods for performance evaluation of both the ensemble method and the selected ensemble members on RS126 and CB513 datasets. The methods used are Cross-Validation (section 2.3), Confusion Matrix and three accuracy measurements. A description of these methods will be given in the following subsections.

Confusion matrix

In order to compute the statistics, a matrix of size 3 × 3 named confusion matrix (contingency table) has been used, $M = (M_{ij})_{1 \le i, j \le 3}$, where M_{ij} denotes the number of residues observed in state i and predicted in state j. The rows indicate the states of the secondary structure obtained by DSSP and the columns show the states of the secondary structure predicted by the classifier. The number of correctly predicted examples is the sum of diagonal elements in the matrix, all others are incorrectly predicted.

Prediction accuracy measurements

Several standard accuracy measures have been suggested in the literature. The most popular measure is the three-state overall per-residue accuracy known as Q_3.[7,57] Complementary measures such as the Matthews correlation coefficients (C_H, C_E, C_C)[58] and the segment overlap SOV[9,59] are also currently calculated to evaluate the performance. Here it will be more concise to use these three measures to estimate and compare the performance of the proposed system.

- Per-residue prediction accuracy Q_3: gives percentage of residues whose structural class is correctly predicted. It is obtained as:

  $$Q_3 = 100 \times \frac{1}{N} \sum_{i=1}^{3} M_{ii} \qquad (15)$$

  For each type of secondary structure the per-residue accuracy can also be calculated, for example:

  $$Q_H = 100 \times \frac{1}{n_H} M_{ii} \qquad (16)$$

  with n_H residues observed in helix state. N represents the total number of residues in the sequence.
- The Pearson's Matthews correlation coefficients for a particular state $i \in \{C, E, H\}$ it is given by

  $$C_i = \frac{p_i n_i - u_i o_i}{\sqrt{(p_i + u_i)(p_i + o_i)(n_i + u_i)(n_i + o_i)}} \qquad (17)$$

  With $p_i = M_{ii}$ the number of correctly predicted residues in state i, $n_i = \sum_{j}^{3} \sum_{k \ne i}^{3} M_{jk}$ the number of residues that were correctly rejected (true negatives), $o_i = \sum_{j \ne i}^{3} M_{ij}$ the number of residues that were incorrectly predicted to be in state i (false positives), $u_i = \sum_{j \ne i}^{3} M_{ij}$ the number of residues that were incorrectly rejected (false negatives). The result is a number between −1 and 1, where the value 1 corresponds to a perfect prediction, −1 indicates a complete disagreement and 0 indicates that the prediction has no correlation with the observed structure.

- The Segment OVerlap measure SOV

  Proposed firstly by Rost et al[9] and modified by Zemla et al,[59] it is based on the average overlap between the observed and the predicted segments.

$$SOV_H = \frac{1}{n_H} \sum_{S_H} \frac{minOV\,(S_1,S_2)+\delta(S_1,S_2)}{maxOV\,(S_1,S_2)} \quad (18)$$

  S_1 and S_2 are the observed and predicted secondary structure segments in the helix state, S_H is the number of all ovelapping segment pairs ($S_{1,}$ S_2), *minOV* ($S_{1,}$ S_2) is the length of the actual overlap of S_1 and S_2 and *maxOV* (S_1, S_2) is the length of total extent for which either of the segments S_1 or S_2 has a residue in helix state, n_H is the total number of amino acid residues observed in the helix conformation. The definition of $\delta(S_{1,}$ $S_2)$ is as follows:

$$\delta(S_1,S_2) = \left\{ \begin{matrix} maxOV\,(S_1,S_2) - minOV\,(S_1,S_2) \\ minOV\,(S_1,S_2) \\ int\,(0.5 \times len\,(S_1)) \\ int\,(0.5 \times len\,(S_2)) \end{matrix} \right\} \quad (19)$$

  where, *len*(S_i) is the number of amino acid residues in the segment S_i. The SOV for all three states is given by;

$$SOV(\%) = \frac{1}{n_{res}} \left(\sum_{i \in H,E,C} \sum_{S(i)} \left[\frac{minOV\,(S_1,S_2)+\delta\,(S_1,S_2)}{maxOV\,(S_1,S_2)} \times len\,(S_1) \right] \right) \times 100 \quad (20)$$

  Here, S_1 and S_2 are the observed and predicted secondary structure segments in state i and S(i) is the number of all ovelapping segment pairs (S_1, S_2) in state i.

Experimental Results and Discussion

In this section, we report the results of our experiments obtained using the seven-fold cross validation on both RS126 and CB513 datasets. We first evaluate the six individual classifiers with and without refining their outputs by applying the proposed filter and then try to evaluate whether and how much voting can improve the prediction accuracy. Morover, a comparison is made between the three voting schemes with and without filtering the predictions. The experiments have been conducted in C on linux platform. For the M-SVMs, the use of a proper kernel function with its optimal parameters can achieve high prediction accuracy. However, there are no effective theoretical methods for model selection and the optimization of kernel parameters may become quite diffcult due to computing time. In this study, we used the gaussian radial basis function (RBF) kernel for both C-SVC, the M-SVM of Weston and Watkins (M-SVM$_{WW}$) and the M-SVM of Crammer and Singer (M-SVM$_{CS}$). This kernel choice is motivated by the fact that the RBF kernel is more suitable for complex classification problems. The parameters γ and *C* have been used after trials. Different γ and *C* pairs have been tested to find out the optimum parameter values. Finally, for C-SVC the parameter of the kernel function $\gamma = 0.0031$, the regularization parameter $C = 1.0$ were adopted for the RS126 and $\gamma = 0.0015$ and $C = 10$ for CB513. For the two M-SVMs, we used $C = 1.0$. Also, it remains difficult to determine appropriate structure of the Artificial Neural Networks (ANNs), so the MLP has been experimented using a single hidden layer with 10 units which has been found to be the most effective number in our training stage. QuickRBF is used with 12000 selected centers. The performance of all classfiers has been evaluated using the Q_3, SOV and Matthews correlation coefficients ($C_{H/E/C}$).

Prediction accuracies of the six individual classifiers

The six selected classifiers have been evaluated individually so as to estimate their performance in predicting the three conformational states. Table 1 shows that the best individual classifier for RS126 dataset is M-SVM$_{WW}$. QuickRBF and M-SVM$_{CS}$ also achieve good prediction accuracy. The results show that C-SVC and k-NN are here poorer performing classifiers in the ensemble. Table 2 reports the eventual performance that we can obtain if we filter each individual classifier outputs.

Table 1. Performance comparison of the six individual classifiers for RS126 dataset.

Classifiers	Accuracy measures										
	Q_3(%)	Q_H(%)	Q_E(%)	Q_C(%)	C_H	C_E	C_C	SOV_H(%)	SOV_E(%)	SOV_C(%)	SOV(%)
M-SVM$_{WW}$	78.11	77.36	65.80	84.46	0.724	0.624	0.610	73.0	65.7	70.2	70.1
M-SVM$_{CS}$	77.79	77.17	64.85	84.35	0.710	0.621	0.611	72.9	66.0	71.4	70.9
C-SVC	72.58	66.91	55.30	84.55	0.630	0.522	0.534	64.0	61.1	66.5	64.1
RBFNN	77.05	75.92	62.49	84.72	0.709	0.601	0.595	71.0	62.0	68.1	67.1
MLP	74.42	73.25	61.24	81.47	0.662	0.554	0.562	64.0	60.9	63.9	62.7
k-NN	72.69	66.88	50.00	87.33	0.605	0.521	0.557	64.0	53.2	98.2	63.2

The results obtained on CB513 dataset are reported in Tables 3 and 4. Both QuickRBF, MSVM$_{WW}$ and M-SVM$_{CS}$ achieve good performance. C-SVC achieves better accuracy for CB513 than RS126. The poorer-performed classifiers are the k-NN and the MLP. Table 4 shows that after filtering the outputs, the Q_3 and SOV scores increased significantly.

Generally, coils are predicted quite well, helices are predicted moderately well and strands are predicted rather poorly. Our analysis shows the same trend. All the results on both RS126 and CB513 confirmed this well. It can be clearly seen that the coil predictions have a high prediction accuracy compared to the β-sheet as well as the α-helix. The results show also that each classifier has its particular stengths and weaknesses in predicting structural conformations. The high number of predicted coils here is mainly due to the fact that shorter secondary structure elements are harder to predict.

Prediction accuracies of the ensemble method using the three voting schemes

Tables 5 and 7 report the results obtained by the three voting schemes on RS126 and CB513 without refining the obtained outputs. We can see that Influenced Majority Voting (IMV) gives better results than Simple Majority Voting (SMV) but the results given by Weighted Majority Voting (WMV) are best. By combining the classifiers predictions, the SOV increased significantly, which means that the segments prediction quality becomes better. The best Q_3 score has been achieved by WMV for both RS126 and CB513. The prediction accuracies after applying the filter to the predictions are listed in Tables 6 and 8. A comparison between prediction accuracies for each conformational state in terms of Q_3 and SOV scores is plotted in Figure 1 for RS126 and Figure 2 for CB513.

We can see in Figure 1 and Figure 2 that the $Q_{H|E|C}$ and $SOV_{H|E|C}$ scores obtained using the three combination schemes are higher than the scores achieved by the best individual classifier in the ensemble. Furthermore, the scores achieved are even higher than those obtained by refining the predictions of the best individual classifier. We can see also that the coil prediction quality is more pronounced as well as the helix conformation. If WMV is used, the prediction accuracies of coil conformations increase obviously. The gain in prediction quality is then more significant using WMV as majority voting combination scheme rather than SMV.

Table 2. Performance comparison of the six individual classifiers for RS126 dataset after applying the filter to the predictions.

Classifiers + filter	Accuracy measures										
	Q_3(%)	Q_H(%)	Q_E(%)	Q_C(%)	C_H	C_E	C_C	SOV_H(%)	SOV_E(%)	SOV_C(%)	SOV(%)
M-SVM$_{WW}$	78.35	76.56	64.93	85.92	0.735	0.624	0.612	76.9	66.6	71.3	73.4
M-SVM$_{CS}$	77.94	76.49	63.75	85.64	0.719	0.619	0.611	76.0	66.9	72.2	73.7
C-SVC	72.67	66.12	54.24	85.78	0.642	0521	0.530	67.0	61.9	67.4	67.5
RBFNN	77.21	74.78	62.16	85.98	0.718	0.604	0.593	75.0	65.8	70.6	72.3
MLP	74.83	72.50	60.68	83.11	0.678	0.555	0.564	71.9	63.6	68.4	70.1
k-NN	72.99	66.57	48.03	89.10	0.633	0.518	0.551	71.3	52.9	67.4	66.7

Table 3. Performance comparison of the six individual classifiers for CB513 dataset.

Classifiers	Accuracy measures										
	Q_3(%)	Q_H(%)	Q_E(%)	Q_C(%)	C_H	C_E	C_C	SOV_H(%)	SOV_E(%)	SOV_C(%)	SOV(%)
M-SVM$_{WW}$	76.08	76.95	64.48	81.51	0.704	0.614	0.568	71.4	65.1	67.9	69.7
M-SVM$_{CS}$	76.11	76.97	64.96	81.33	0.706	0.614	0.568	71.0	65.3	67.8	69.5
C-SVC	73.32	72.31	55.78	83.43	0.673	0.564	0.523	67.3	58.1	64.8	65.2
RBFNN	76.04	77.39	62.46	82.15	0.700	0.611	0.572	70.7	64.8	69.1	70.1
MLP	72.97	74.15	62.09	77.77	0.645	0.560	0.532	62.8	62.5	64.7	63.3
k-NN	72.82	81.06	57.29	74.39	0.642	0.549	0.536	72.4	59.6	64.0	66.9

All the experimental results reported above show that the proposed system exploits the prediction power of the selected classifiers but it is also influenced by the worst classifiers. In all the cases, the proppposed system works more effectively than the individual classifiers. In this study, we have presented a real case to demonstrate the advantage of combining classifiers. Our results, as shown in all the Tables, demonstrate substantially higher accuracy achieved by M-SVMs as compared to MLP, k-NN and C-SVC. The experiments provided evidence from the several tests conducted that combining highly accurate classifiers improve significantly the performance and that majority voting rule is a robust way to combine powerful classifiers. Figure 3 summarizes the performances of the individual classifiers and the three schemes of the ensemble method proposed with and without refining the predictions.

The aim of combining multiple classifiers is to obtain better performance than individual classifier. Here, the experimental evaluation using the two standard datasets RS126 and CB513 show that the proposed framework indeed improve the prediction quality. The results reported here are highly encouraging since they reveal that the proposed framework is capable of producing higher Q_3 and SOV scores than that achieved by PHD, DSC, PREDATOR, NNSSP and Jpred as well as previously developed SVM-based methods and similar to the current prominent PSSP methods such as PSIPRED, SSpro, SAM-T99sec. It is difficult to compare exactly our results against the results reported in recently published studies directly because it will not be a fair comparison to compare only the absolute Q_3 values, since different DSSP assignment schemes and different training sets and test proteins are used. However, further efforts must be made to design efficient combination system to obtain finer predictions of structural conformations, especially of β-sheet structures which remain the most difficult to predict. The evidence clearly supports the conclusion that combining classifiers is of benefit to this problem. More tests with larger training and test datasets using different DSSP assignment schemes and including further M-SVMs are in progress to obtain a more statistically increase in prediction accuracy. It should also be noted that there is scope for further improvement of the post-processing filter.

Table 4. Performance comparison of the six individual classifiers for CB513 dataset after applaying the filter to the predicitions.

Classifiers + filter	Accuracy measures										
	Q_3(%)	Q_H(%)	Q_E(%)	Q_C(%)	C_H	C_E	C_C	SOV_H(%)	SOV_E(%)	SOV_C(%)	SOV(%)
M-SVM$_{WW}$	76.34	76.13	64.36	82.85	0.714	0.619	0.569	73.1	66.5	69.4	72.7
M-SVM$_{CS}$	76.34	76.16	64.79	82.61	0.715	0.618	0.569	73.1	66.6	69.4	72.8
C-SVC	73.55	71.35	55.34	84.97	0.681	0568	0.526	70.3	59.2	65.5	68.2
RBFNN	76.30	76.56	62.04	83.64	0.710	0.615	0.573	73.1	65.9	70.0	72.9
MLP	73.69	73.87	62.19	79.63	0.666	0.570	0.536	71.0	65.4	67.9	70.7
k-NN	73.18	80.40	56.74	76.05	0.656	0.551	0.533	73.5	60.8	66.6	70.1

Table 5. Performance comparison of the three combination schemes for RS126 dataset.

Ensemble method		Accuracy measures										
		Q_3(%)	Q_H(%)	Q_E(%)	Q_C(%)	C_H	C_E	C_C	SOV_H(%)	SOV_E(%)	SOV_C(%)	SOV(%)
SMV	M-SVM$_{WW}$, M-SVM$_{CS}$	77.84	79.07	66.63	82.36	0.711	0.625	0.612	74.0	66.5	70.1	70.7
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF	78.18	77.38	65.10	84.93	0.722	0.625	0.615	72.7	66.0	71.2	70.7
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP	77.94	78.72	66.51	82.85	0.716	0.624	0.611	73.7	66.4	69.1	69.9
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC	77.53	76.06	64.02	84.93	0.710	0.614	0.605	71.7	65.4	70.1	69.6
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC, k-NN	78.08	77.59	65.24	84.52	0.715	0.626	0.615	74.0	66.5	70.4	70.8
IMV	M-SVM$_{WW}$, M-SVM$_{CS}$	78.11	77.36	65.80	84.46	0.724	0.624	0.610	73.0	65.7	70.2	70.1
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF	78.15	77.24	65.86	84.60	0.726	0.625	0.611	73.1	65.7	70.8	70.5
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP	78.15	77.31	65.84	84.56	0.726	0.624	0.611	73.2	65.7	70.7	70.5
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC	77.93	76.27	65.84	84.78	0.720	0.624	0.607	73.3	65.9	70.8	70.9
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC, k-NN	78.21	76.78	65.91	85.00	0.726	0.626	0.612	73.9	65.8	71.1	71.1
WMV	M-SVM$_{WW}$, M-SVM$_{CS}$	78.11	77.36	65.80	84.46	0.724	0.624	0.610	73.8	65.7	70.2	70.1
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF	78.24	77.24	65.47	84.98	0.726	0.626	0.614	73.1	66.0	71.7	71.1
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP	78.24	77.27	65.45	84.97	0.726	0.626	0.613	73.2	66.1	71.8	71.1
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC	77.80	76.27	64.39	85.20	0.720	0.616	0.607	73.2	65.5	70.8	70.5
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC, k-NN	78.41	76.72	65.14	85.85	0.726	0.630	0.618	73.8	66.6	72.0	71.8

Table 6. Performance comparison of the three combination schemes after applying the filter to the predictions for RS126 dataset.

Ensemble method + filter		Accuracy measures										
		Q_3(%)	Q_H(%)	Q_E(%)	Q_C(%)	C_H	C_E	C_C	SOV_H(%)	SOV_E(%)	SOV_C(%)	SOV(%)
SMV	M-SVM$_{WW}$, M-SVM$_{CS}$	78.14	78.46	65.91	83.75	0.721	0.626	0.613	77.6	67.7	71.9	74.2
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF	78.29	76.64	63.96	86.20	0.731	0.623	0.613	76.6	66.6	72.0	73.7
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP	78.19	77.89	65.99	84.18	0.727	0.625	0.611	76.8	67.8	71.8	73.9
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC	77.67	75.21	63.32	86.11	0.719	0.615	0.603	75.3	67.2	70.9	72.9
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC, k-NN	78.34	77.03	64.37	85.85	0.728	0.625	0.615	76.8	67.5	71.8	73.8
IMV	M-SVM$_{WW}$, M-SVM$_{CS}$	78.34	77.24	64.93	85.44	0.733	0.624	0.612	77.2	66.6	71.0	73.2
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF	78.32	77.04	64.99	85.50	0.731	0.625	0.612	76.9	66.6	71.4	73.3
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP	78.29	77.06	65.03	85.42	0.732	0.625	0.611	76.7	66.6	71.1	73.1
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC	77.93	75.59	64.93	85.66	0.721	0.625	0.606	75.8	66.7	70.7	72.9
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC, k-NN	78.41	76.36	65.16	86.06	0.731	0.627	0.615	76.7	66.8	71.0	73.3
WMV	M-SVM$_{WW}$, M-SVM$_{CS}$	78.35	76.56	64.93	85.92	0.735	0.624	0.612	76.9	66.6	71.3	73.4
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF	78.25	76.49	64.06	86.17	0.731	0.622	0.612	76.4	66.6	71.9	73.6
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP	78.26	76.52	64.12	86.14	0.731	0.622	0.612	76.5	66.7	72.0	73.7
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC	77.67	75.04	63.32	86.23	0.720	0.615	0.603	75.2	67.2	70.9	72.9
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC, k-NN	78.44	75.85	63.77	87.12	0.731	0.626	0.617	76.3	67.3	72.0	73.7

Table 7. Performance comparison of the three combination schemes for CB513 dataset.

Ensemble method		**Accuracy measures**										
		Q_3(%)	**Q_H(%)**	**Q_E(%)**	**Q_C(%)**	**C_H**	**C_E**	**C_C**	**SOV_H(%)**	**SOV_E(%)**	**SOV_C(%)**	**SOV(%)**
SMV	M-SVM$_{WW}$, M-SVM$_{CS}$	76.34	76.16	64.79	82.61	0.715	0.618	0.569	73.1	66.6	69.4	72.8
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF	76.19	77.0	64.59	81.69	0.706	0.616	0.570	71.4	65.3	68.1	69.9
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP	76.27	78.64	65.78	79.92	0.706	0.618	0.571	72.5	66.0	67.6	70.2
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC	76.34	77.82	63.51	82.58	0.709	0.615	0.575	71.8	64.9	68.9	70.4
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC, k-NN	76.39	78.64	64.18	81.03	0.706	0.618	0.575	73.1	65.5	68.5	70.7
IMV	M-SVM$_{WW}$, M-SVM$_{CS}$	76.11	76.97	64.96	81.33	0.706	0.614	0.568	71.0	65.3	67.8	69.5
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF	76.15	76.92	64.97	81.45	0.707	0.614	0.569	71.4	65.3	67.8	69.8
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP	76.19	77.47	62.99	82.14	0.705	0.612	0.573	71.8	65.2	69.4	70.9
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC	76.27	76.73	65.02	81.86	0.710	0.615	0.571	71.7	65.4	68.4	70.2
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC, k-NN	76.26	76.94	64.93	81.71	0.709	0.616	0.571	71.7	65.4	68.3	70.1
WMV	M-SVM$_{WW}$, M-SVM$_{CS}$	76.11	76.97	64.96	81.33	0.706	0.614	0.568	71.0	65.3	67.8	69.5
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF	76.20	76.92	64.68	81.71	0.707	0.615	0.570	71.4	65.3	68.2	70.0
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP	76.21	76.92	64.70	81.73	0.707	0.615	0.570	71.4	65.3	68.2	70.1
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC	76.33	76.71	63.74	82.68	0.710	0.614	0.574	71.8	64.9	69.1	70.5
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC, k-NN	76.41	77.58	63.98	82.04	0.708	0.618	0.575	72.5	66.0	69.2	71.2

Table 8. Performance comparison of the three combination schemes after applying the filter to the predictions for CB513 dataset.

Ensemble method + filter		Accuracy measures										
		Q_3(%)	Q_H(%)	Q_E(%)	Q_C(%)	C_H	C_E	C_C	SOV_H(%)	SOV_E(%)	SOV_C(%)	SOV(%)
SMV	M-SVM$_{WW}$, M-SVM$_{CS}$	76.65	77.80	63.87	82.49	0.715	0.623	0.576	74.1	66.6	69.9	73.3
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF	76.59	77.46	64.35	82.36	0.715	0.622	0.574	73.9	66.7	70.1	73.3
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP	76.54	77.78	65.62	81.32	0.715	0.622	0.572	74.0	67.3	69.6	73.3
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC	76.59	77.14	63.54	83.05	0.715	0.621	0.575	73.4	66.6	69.9	73.2
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC, k-NN	76.65	77.80	63.87	82.49	0.715	0.623	0.576	74.1	66.6	69.9	73.3
IMV	M-SVM$_{WW}$, M-SVM$_{CS}$	76.33	76.75	64.79	82.10	0.713	0.618	0.569	73.3	66.6	69.3	72.7
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF	76.33	76.62	64.82	82.17	0.712	0.618	0.569	73.1	66.6	69.4	72.7
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP	76.43	77.42	62.56	82.96	0.712	0.615	0.575	73.6	66.1	70.2	73.1
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC	76.51	78.17	62.61	82.53	0.713	0.616	0.576	73.9	66.2	70.1	73.2
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, C-SVC, k-NN	76.51	76.82	64.69	82.52	0.715	0.620	0.573	73.4	66.5	69.7	73.0
WMV	M-SVM$_{WW}$, M-SVM$_{CS}$	76.41	77.58	63.98	82.04	0.708	0.618	0.575	72.5	66.0	69.2	71.2
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF	76.59	77.32	64.48	82.41	0.716	0.622	0.574	73.8	66.8	70.1	73.3
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP	76.57	77.32	64.48	82.38	0.715	0.621	0.574	73.8	66.8	70.1	73.3
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, k-NN	76.64	77.92	63.61	82.51	0.716	0.621	0.576	73.9	66.7	70.1	73.4
	M-SVM$_{WW}$, M-SVM$_{CS}$, RBF, MLP, k-NN, C-SVC	76.69	77.40	63.59	83.06	0.717	0.622	0.577	74.1	66.5	70.3	73.4

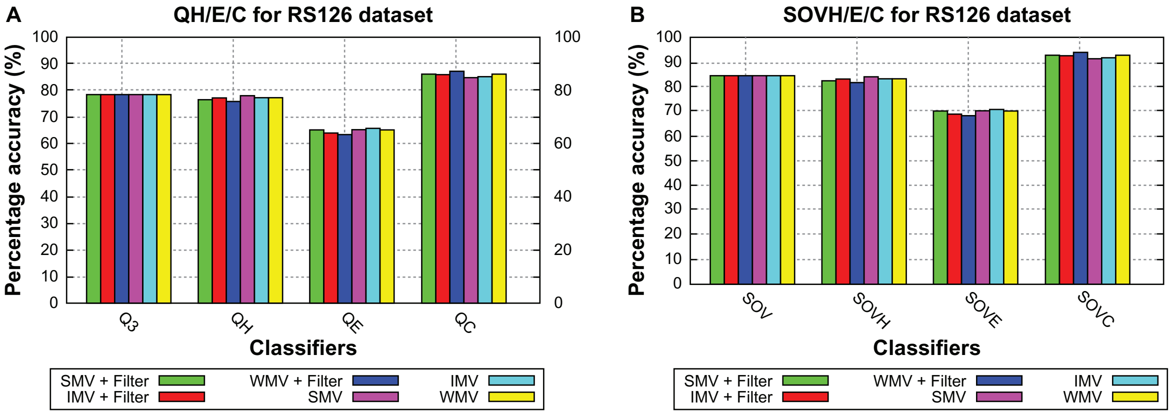

Figure 1. The Q3 (**A**) and SOV (**B**) scores for the three voting schemes on the RS126 dataset. QH/E/C and SOVH/E/C are respectively the predicted Q and SOV scores for each conformational state (helix, strand and coil).

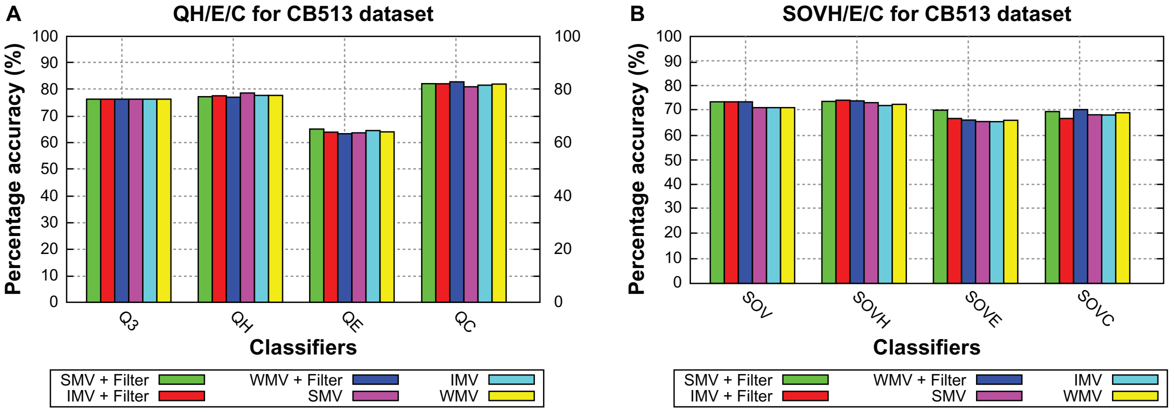

Figure 2. The Q3 (**A**) and SOV (**B**) scores for the three voting schemes on the CB513 dataset. QH/E/C and SOVH/E/C are respectively the predicted Q and SOV scores for each conformational state (helix, strand and coil).

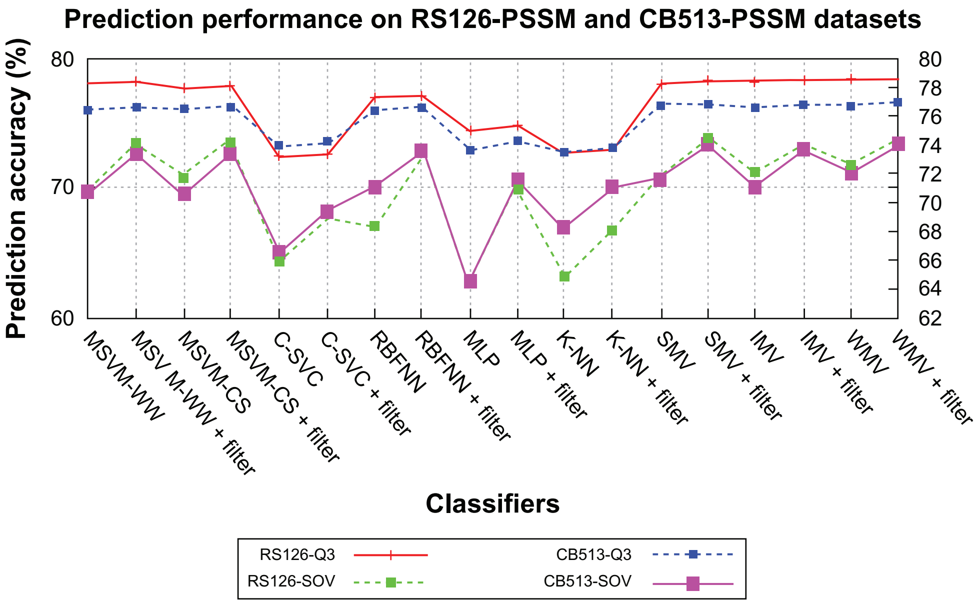

Figure 3. Comparison of prediction accuracies (y axis) between the three combination schemes and the individual classifiers (x axis) with and without applying the filter to the predictions on both RS126 and CB513 datasets.

Conclusion

Ensemble classifier combination has been extensively studied in the last decade, and has been shown to be successful in improving the performance of diverse applications. Protein secondary structure prediction is an important step towards predicting the tertiary structure of proteins and then their function. Therefore, an accurate prediction is strongly required. The effectiveness of the methods used depends crucially on the parameters. A problem which occurs for this study is the selection of ideal parameters for each predictor. We believe that at least three optimization processes are mainly important to perform in PSSP for prediction quality such as for example encoding scheme, sliding window size and parameter optimization. The interesting point emerging from our study is that when multiple classifiers are combined, the gain in performance is not always guaranteed, especially when poor-performed classifiers are integrated. The voting results shows that when lower-quality classifiers are added in, the better classifiers are steadily drowned out. So the gain in performance will be more pronounced by including better-performed classifiers, while including poor-performed classifiers decrease the prediction accuracy. Consequently, the relative merits of maintaining the best and eliminating the weakest can be considered. This study also shows that the SVMs remain the major competitor of ANNs in the field of machine learning. The prediction result shows that *β—sheets* were predicted much more poorly than helices or coils. Generally, coils are the easiest to predict, while *β—sheets* with their long range interactions are the most difficult. Nevertheless, additional experiments can be performed in order to enhance the strengths of the integrated classifiers and to merge other relevant classifiers using skew datasets so as to improve the accuracy of the predictions. The PSSP methods are now relatively mature but despite this progress, much remains to be done. Further improvement is still needed by in silico PSSP methods to reach the precision provided by the experimental techniques. In future research, additional information will be included, such as amino acid physiochemical properties, to investigate possible improvements to our approach, which will be evaluated on more proteins collected from the PDB SELECT database.

Abbreviations

ANN, Artifial Neural Network; BLAST, Basic Local Alignment Search Tool; BLOSUM, BLOck SUbstitution Matrix; GOR, Garnier-Osguthorpe-Robson; MLP, Multi-Layer Perceptron; M-SVM, Multi-class Support Vector Machines; IMV, Influenced Majority Voting; PSI-BLAST, Position Specifi Iterative BLAST; RBFNN, Radial Basis Function Neural Network; SMV, Simple Majority Voting; SVM, Support Vector Machines; WMV, Weighted Majority Voting.

Disclosures

Author(s) have provided signed confirmations to the publisher of their compliance with all applicable legal and ethical obligations in respect to declaration of conflicts of interest, funding, authorship and contributorship, and compliance with ethical requirements in respect to treatment of human and animal test subjects. If this article contains identifiable human subject(s) author(s) were required to supply signed patient consent prior to publication. Author(s) have confirmed that the published article is unique and not under consideration nor published by any other publication and that they have consent to reproduce any copyrighted material. The peer reviewers declared no conflicts of interest.

References

1. Anfinsen C. Principles that govern the folding of protein chains. *Science*. 1973;181:223.
2. Lim V. Structural Principles of the Globular Organization of Protein Chains. A Stereochemical Theory of Globular Protein Secondary Structure. *Journal of Molecular Biology*. 1974;88(4):857–72.
3. Chou P, Fasman G. Empirical predictions of protein conformation. *Annual Review of Bio-chemistry*. 1978;47:251–76.
4. Garnier J, Robson DJOB. Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of Molecular Biology*. 1978;120:97–120.
5. Gibrat J, Garnier J, Robson B. Further developments of protein secondary structure prediction using information theory. *Journal of Molecular Biology*. 1987;198:425–43.
6. Biou V, Gibrat J, Levin J, Robson B, Garnier J. Secondary structure prediction: combination of three different methods. *Protein Ingineering*. 1988;2:185–91.
7. Qian N, Sejnowski T. Predicting the secondary structure of globular proteins using neural network models. *Journal of Molecular Biology*. 1988;202:865–84.
8. Rost B, Sander C. Prediction of secondary structure at better than 70% accuracy. *Journal of Molecular Biology*. 1993;232:584–99.
9. Rost B, Sander C. Combining evolutionnary information and neural networks to predict protein secondary structure prediction. *Proteins*. 1994;19:55–72.
10. Geourjon C, Deléage G. SOPM: a self-optimized method for protein secondary structure prediction. *Protein Engineering*. 1994;7(2):157–64.

[g] Subset of the structures in the PDB that does not contain (highly) homolog sequences availible at http://swift.cmbi.kun.nl/swift/pdbsel/.

[h] Structural Classification of Proteins availible at http://scop.mrc-lmb.cam.ac.uk/scop/.

11. King RD, Sternberg MJE. Identification and application of the concepts important for accurate and reliable protein secondary structure prediction. *Protein Science*. 1996;5:2298–310.
12. Salamov AA, Solovyev VV. Prediction of protein secondary structure by combining nearest neighbor algorithms and multiple sequence alignments. *Journal of Molecular Biology*. 1995;247:11–5.
13. Frishman D, Argos P. Incorporation of non-local interactions in protein secondary structure prediction from the amino-acid sequence. *Protein Engineering*. 1996;9:133–42.
14. Riis S, Krogh A. Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments. *Journal of Computational Biology*. 1996;3:163–83.
15. Jones D. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*. 1999;292:195–202.
16. Cuff JA, Barton G. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure Function and Genetics*. 1999;34(4):508–19.
17. Cheng J, Randall A, Sweredoski M, Baldi P. SCRATCH: a Protein Structure and Structural Feature Prediction Server. *Nucleic Acids Research*. 2005;33:72–6.
18. Pollastri G, McLysaght A. Porter: a new, accurate server for protein secondary structure prediction. *Bioinformatics*. 2005;21(8):1719–20.
19. Bondugula R, Xu D. MUPRED: A Tool for Bridging the Gap between Template Based Methods and Sequence Profile Based Methods for Protein Secondary Structure Prediction. *Proteins*. 2007;66(3):664–70.
20. Karplus K, Barrett C, Hughey R. Hidden Markov models for detecting remote ptotein homologies. *Bioinformatics*. 1998;14:846–56.
21. Petersen TN, Lundegaard C, Nielsen M, et al. Prediction of protein secondary structure at 80% accuracy. *Proteins*. 2000;41:17–20.
22. Hua S, Sun Z. A novel method of protein secondary structure prediction with high segment overlap measure: Support vector machine approach. *Journal of Molecular Biology*. 2001;308:397–407.
23. Hsu CW, Lin CJ. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*. 2002;13(2): 415–25.
24. Kim H, Park H. Protein secondary structure prediction based on an improved support vector machines approach. *Protein Engineering*. 2003;16(8):553–60.
25. Nguyen M, Rajapkse J. Multi-Class Support Vector Machines for Secondary Structure Prediction. *Genome Informatics*. 2003;14:218–27.
26. Nguyen M, Rajapkse J. Two-stage multi-class support vector machines to protein secondary structure prediction. *Pac Symp Biocomput*. 2005;10: 346–57.
27. Ward J, Gufin L, Buxton B, Jones D. Secondary structure prediction with support vector machines. *Bioinformatics*. 2003;19:1650–5.
28. Guo J, Chen H, Sun Z, Lin Y. A Novel Method for Protein Secondary Structure Prediction Using Dual-Layer SVM and Profiles. *Proteins: Structure, Function, and Bioinformatics*. 2004;54:738–43.
29. Hu H, Harrison R, Tai P, Pan Y. Knowledge discovery in bioinformatics. In Hu X, Pan Y, editors. *Techniques, Methods, and Applications,* ch. 1, 1–26, John Wiley Sons, Inc; 2007.
30. Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen bonded and geometrical features. *Biopolymers*. 1983;22:2577–637.
31. Frishman D, Argos P. Knowledge-based protein secondary structure assignment. *Proteins*. 1995;23:566–79.
32. Richards F, Kundrot C. Identification of structural motifs from protein coordinate data: secondary structure and first level super secondary structure. *Proteins*. 1988;3:71–84.
33. Martin J, Letellier G, Marin A, Brevern A, Gibrat G. Protein secondary structure assignment revisited: a datailed analysis of different assignment methods. *BMC Struct Biol*. 2005;5:17.
34. Altschul S, Madden T, Schäffer A, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*. 1997;25(17):3389–402.
35. Eddy SR. Profile Hidden Markov models. *Bioinformatics*. 1998;14: 755–63.
36. Heniko S, Henikoff J. Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A*. 1992;89:10915–9.
37. Chang C, Lin C. LIBSVM: a library for support vector machines. *SIAM J Appl Math*. 2001. Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm.
38. Paredes R, Yi D, Vidal E. Learning weighted metrics to minimize nearest-neighbor classifivation error. *IEEE Transaction on Pattern Analysis and Machine Intelligence*. 2006;28(7):1100–10.
39. Minsky M, Papert S. Perceptron: an essay in computational geometry. *MIT Press*. 1969.
40. Narayan S. The generalized sigmoid activation function: competitive supervised learning. *Information Sciences*. 1997;99(1–2):69–82.
41. Rumellart DE, Hinton GE Williams RJ. Learning internal representation by errors propagation. *MIT Press Cambridge*. 1986;1(1–2):318–62.
42. Moody J, Darken CJ. Fast learning in networks of locally-tuned processing units. *Neural Computation*. 1989;1:281–94.
43. Poggio T, Girosi T. Networks for approximation and learning. *Proc IEEE*. 1990;78:1481–97.
44. Ou Y, Oyang Y, Chen C. A novel radial basis function network classifier with centers set by hierarchical clustering. *International Joint Conference on Neural Networks (IJCNN)*. 2005:1.
45. Boser B, Guyon I, Vapnik V. A training algorithm for optimal margin classifiers. *COLT'92*. 1992:144–52.
46. Cortes C, Vapnik V. Support-vector networks. *Machine Learning*. 1995; 20(3):273–97.
47. Vapnik V. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, New York; 1982.
48. Aizerman A, Braverman E, Rozone L. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*. 1964;25:821–37.
49. Vapnik V. *Statistical Learning Theory*. John Wiley & Sons, Inc., New York; 1998.
50. Schölkopf B, Burges C, Smola A. editors. *Advances in Kernel Methods, Support Vector Learning*. The MIT Press; 1999.
51. Rifkin R, Klautau A. In defense of one-vs-all classification. *Journal of Machine Learning Research*. 2004;5:101–41.
52. Platt J, Cristianini N, Shawe-Taylor J. Large margin DAGs for multiclass classification. *NIPS 12*. 2000:547–53.
53. Dietterich T, Bakiri G. Error-correcting output codes: A general method for improving multiclass inductive learning programs. *Ninth National Conference on Articial Intelligence (AAAI-91)*. 1991:572–7.
54. Weston J, Watkins C. "Multi-class support vector machines," Tech. Rep. CSD-TR-98-04, Royal Holloway, University of London, Department of Computer Science; 1998.
55. Crammer K, Singer Y. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*. 2001;2:265–92.
56. King R, Sternberg M. Identification application of the concepts important for the accurate and reliable protein secondary structure prediction. *Protein Science*. 1996;5:2298–310.
57. Rost B, Sander. C. Prediction of protein secondary structure at better than 70% accuracy. *Journal of Molecular Biology*. 1993;232(2): 584–99.
58. Matthews B. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica and Biophysica Acta*. 1975;405: 442–51.
59. Zemla A, Venclovas C, Fidelis K, Rost B. A Modified Definition of Sov, a Segment-Based Measure for Protein Secondary Structure Prediction Assessment. *Proteins: Structure, Function and Genetics*. 1999;34: 220–3.