Protein secondary structure prediction is one of the most important and challenging problems in bioinformatics. Machine learning techniques have been applied to solve the problem and have gained substantial success in this research area. However, there is still room for improvement toward the theoretical limit. In this paper, we present a novel method for protein secondary structure prediction based on a data partition and semi-random subspace method (PSRSM). Data partitioning is an important strategy for our method. First, the protein training dataset was partitioned into several subsets based on the length of the protein sequence. Then we trained base classifiers on the subspace data generated by the semi-random subspace method, and combined base classifiers by majority vote rule into ensemble classifiers on each subset. Multiple classifiers were trained on different subsets. These different classifiers were used to predict the secondary structures of different proteins according to the protein sequence length. Experiments are performed on 2SPDB, CB513, CASP10, CASP11, CASP12, and T100 datasets, and the good performance of 86.38%, 84.53%, 85.51%, 85.89%, 85.55%, and 85.09% is achieved respectively. Experimental results showed that our method outperforms other state-of-the-art methods.

Proteins play a key role in almost all biological processes; they are the basis of life. For example, they take part in maintaining the structural integrity of the cell, transport and storage of small molecules, catalysis, regulation, signaling, and the immune system. There are 20 different amino acids that form proteins in nature. The amino acids of a protein are connected in sequence with the carboxyl group of one amino acid forming a peptide bond with the amino group of the next amino acid. Protein structure is essential for the understanding of protein function. In order to recognize the protein functions of proteins at a molecular level, it is sometimes necessary to determine their 3D structure. Accurately and reliably predicting structures from protein sequences is one of the most challenging tasks in computational biology. Protein secondary structure prediction provides a significant first step toward tertiary structure prediction, as well as offering information about protein activity, relationships, and functions.

Protein secondary structure refers to the local conformation proteins’ polypeptide backbone. There are two regular secondary structure states, $\alpha$-helix (H) and $\beta$-strand (E), and one irregular secondary structure type, the coil region (C). Sander developed a secondary structure assignment method Dictionary of Secondary Structure of Proteins (DSSP), which automatically assigns secondary structure into eight states (H, E, B, T, S, L, G, and I) according to hydrogen-bonding patterns. These eight states are often further simplified into three states of helix, sheet, and coil. The most widely used convention is that helix is designated as G, H and I; sheet as B and E; and all other states are designated as a coils. Most commonly, the secondary structure prediction problem is formulated as follows: given a protein sequence with amino acids, predict whether each amino acid is in the $\alpha$-helix (H), $\beta$-strand (E), or coil region (C). Protein secondary structure prediction is usually evaluated by Q3 accuracy, which measures the percentage of residues for three-state secondary structures to determine whether they have been predicted correctly.

Protein secondary structure prediction began in 1951 when Pauling and Corey predicted helical and sheet conformations for protein polypeptide backbones, even before the first protein structure was determined.
Many statistical approaches and machine learning approaches have been developed to predict secondary structure. One of the first approaches for predicting protein secondary structure, uses a combination of statistical and heuristic rules\(^4,5\). The GOR\(^6\) method formalizes the secondary structure prediction problem within an information-theoretic framework. Position specific scoring matrix (PSSM)\(^7\) based on PSIBLAST\(^8\) reflects evolutionary information and has made the most significant improvements in protein secondary structure prediction. Many machine learning methods have been developed to predict protein secondary structure, and exhibit good performance by exploiting evolutionary information, as well as statistical information about amino acid subsequences\(^9\). For example, many neural network (NN)\(^10\)\(^–\)14 methods, hidden Markov model (HMM)\(^15\)\(^–\)17, support vector machines (SVM)\(^18\)\(^–\)21, and K-nearest neighbors\(^22\) have had substantial success, and Q3 accuracy has reached to 80%. 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\(^23\). Recently, several papers used deep learning networks\(^24\)\(^–\)28 to predict protein secondary structure and obtained good success. The highest Q3 accuracy without relying on structure templates is now at 82\textendash}84\%\(^3\). DeepCNF\(^27\) is a deep learning extension of conditional neural fields (CNF), which integrates conditional random fields and shallow neural networks. The overall performance of DeepCNF is significantly better than other state-of-the-art methods, breaking the long-lasting \textasciitilde80% accuracy. Recently SPIDER3 improved the prediction of protein secondary structure by capturing non-local interactions using long short-term memory bidirectional recurrent neural networks\(^29\). In the paper\(^30\), a new deep inception-inside-inception network, called MUFOLD-SS, was proposed for protein secondary structure prediction. SPIDER3 and MUFOLD-SS achieved better performance, compared to DeepCNF.

In this paper, we presented a data partition and semi-random subspace method (PSRSM) for protein secondary structure prediction. The first step was partitioning the protein training dataset into several subsets based on the lengths of protein sequences. The second step was generating subspaces by the semi-random subspaces method, training base classifiers on the subspaces, and then combining them by majority vote rule on each subset. Fig. 1 demonstrates our PSRSM experimental framework.

![Figure 1. PSRSM framework. Training Data D is partitioned into k subsets D\(_1\), D\(_2\), \ldots, D\(_k\), and S\(_j\) is the jth subspace data of subset D\(_i\); C\(_{ij}\) is a base classifier trained on S\(_j\).](image-url)
vector machines (SVMs) as the base classifier. Support vector machines are a popular machine learning method for classification, regression, and other learning tasks. Compared to other machine learning methods, SVM has the advantages of high performance, absence of local minima, and ability to deal with multidimensional datasets, in which with complex relationships exist among data elements. Support vector machines (SVMs) have had substantial success in protein secondary structure prediction.

Experimental results show that the overall performance of PSRSM was better than the current state-of-the-art methods.

Results

Datasets. We used six publicly available datasets (1) ASTRAL31, (2) CullPDB32, (3) CASP1033, (4) CASP1134, (5) CASP1235, (6) CB51336, and (7) 25PDB37 (8) a dataset T100 developed in-house. ASTRAL, ASTRAL + CullPDB, and T100 datasets are available from supplement files.

In this research, we combined the ASTRAL dataset and CullPDB dataset to be our training dataset, i.e., the ASTRAL + CullPDB dataset. The CullPDB dataset was selected based on the percentage identity cutoff of 25%, the resolution cutoff of 3 angstroms, and the R-factor cutoff of 0.25. There are 12,288 proteins in the CullPDB dataset. ASTRAL dataset had 6,892 proteins, with less than 25% sequence identity. Our training dataset ASTRAL + CullPDB had 15,696 proteins; we removed all duplicated proteins.

Publicly available datasets CASP10, CASP11, CASP12, and CB513, and 25PDB were used to evaluate our method and compared using SPINE-X38, JPRED39, PSIPRED40, and DeepCNF. 99 proteins of the CASP10 dataset, 81 proteins of the CASP11 dataset, and 19 proteins of the CASP12 dataset were selected according to the availability of crystal structure. The CB513 dataset has 513 protein sequences. Any two proteins of CB513 share less than 25% sequence identity with each other. The 25PDB dataset was selected with low sequence similarity of no more than 25%, and has 1673 proteins, consisting of 443 all-α, 443 all-β, 346 α/β and 441 α + β. Note that the number of proteins in these datasets may be different from those reported in other published papers because we only used the available online (http://www.rcsb.org/) or with the PSSM program.

In addition, we randomly downloaded 100 new proteins (T100) released after 1 January 2018 from http://www.rcsb.org/. The dataset (T100) contains 100 proteins with sequence lengths ranging from 18 to 1460. We used T100 to test PSRSM and deepCNF using our online servers and their online server RaptorX-Property which was ranked first in secondary structure prediction.

Because T100 dataset is released after 2018, there is no duplicated proteins with our training dataset. All our training datasets were collected before February 2017.

Performance measures. Several different measures can be used to measure the secondary structure prediction accuracy, the most common being Q3. The Q3 accuracy is defined as the percentage of residues for which the predicted secondary structures are correct, Q3 is calculated as follows:

\[ Q_3 = \frac{N_H + N_E + N_C}{N} \times 100, \]

where, \( N_H \), \( N_E \), and \( N_C \), are the number of correctly predicted secondary structures: helix, strand and coil, respectively. \( N \) is the total number of residues (amino acids).

We calculate the average accuracy of the whole test dataset and use average Q3 to evaluate the performance of our model on the test dataset, the average Q3 is defined as

\[ \text{Average Q3} = \frac{\sum_{i=1}^{n} Q_3(X_i)}{n} \]

Where \( n \) is the number of protein sequences that has the valid predicted results in the test dataset, \( X_i \) denotes a protein sequence, and \( Q_3(X_i) \) is the Q3 accuracy of \( X_i \).

Performance. We used Q3 accuracy to compare our PSRSM method with other state-of-the-art methods, SPINE-X, PSIPRED, JPRED, and DeepCNF, on four publicly available datasets (CASP10, CASP11, CASP12, and CB513). Table 1 shows the Q3 accuracy of PSRSM and the other state-of-the-art methods on the four datasets. The experimental results show that PSRSM is significantly outperforming SPINE-X, PSIPRED, and JPRED. Moreover, PSRSM had 1–3% higher Q3 accuracy than DeepCNF. We also tested our method on 25PDB dataset with 1674 proteins, and Q3 accuracy is 86.38%.

In addition, we compared our proposed method to DeepCNF using our online servers (http://210.44.144.20:82/protein_PSRSM/default.aspx) and their online server RaptorX-Property (http://raptorx.uchicago.edu/StructurePropertyPred/predict/) on T100 dataset. Table 2 lists the Q3 accuracy of PSRSM and DeepCNF for each protein. The average Q3 accuracy of PSRSM was higher 2.5% than that of DeepCNF. In addition, we analyzed Q3 accuracy of predicted secondary structures in internal regions and at boundaries. Here, we defined a helical/sheet residue as internal if its two nearest neighboring residues were also helical/sheet residues; we defined it as a boundary if one or both of the nearest neighbors had a different secondary structural assignment. The overall Q3 accuracies of PSRSM and DeepCNF, respectively, were 89.89% and 85.68% in internal regions, and 75.33% and 73.30% at boundaries. We also compared our method with other state-of-the-art methods (SPIDER3, MUFOLD, PSIPRED and JPRED) using their online server on T100 dataset in Table 3. The newly updated MUFOLD and SPIDER3 obtained 89.28% and 88.25% in internal regions, and 74.65% and 70.72% at boundaries. We can see that PSRSM was superior to current state-of-the-art methods not only in internal regions, but also at boundaries.
Discussion

Reason for partitioning training datasets according to protein length rather than randomly.

Our training data was the ASTRAL+CullPDB dataset, which had 15,696 proteins, and 3,863,231 amino acids (AAs). Since training support vector machines on such a large dataset is a very slow process, the first step of our method was partitioning the training data into several different subsets and training SVMs in parallel. If we partitioned the training data randomly, it would just reduce the computation time, but not increase the prediction accuracy. The length of a protein sequence is the number of amino acids in a protein sequence. Protein length is an important feature of a protein because it influences protein structure. For example, the short sequence ‘VVDALVR’ formed ‘EEEEEE’ in six proteins: 1by5_A, 1qfg_A, 1qff_A, 1fcp_A, 1f1l_A, and 2fcp_A. Their lengths are 714, 725, 725, 705, 707, and 723 respectively. Meanwhile ‘VVDALVR’ formed ‘HHHHHH’ in one protein (3vtz_A), and its length was 269. This data can be downloaded at prodata.swmed.edu/chseq. Identical lengths are 714, 725, 725, 705, 707, and 723 respectively. Meanwhile ‘VVDALVR’ formed ‘HHHHHH’ in one protein (3vtz_A), and its length was 269. This data can be downloaded at prodata.swmed.edu/chseq. Identical amino acid sequence has different types of secondary structures in proteins of different lengths; this is because protein length can affect both local and long-range interactions of the protein. Based on the above considerations, we partitioned training datasets according to protein length to cluster proteins in the training data.

In order to validate the effectiveness of our data partitioning strategy, we conducted another experiment. We randomly generated a subset of the ASTRAL+CullPDB dataset randomly instead of according to protein length, and similarly trained SVM base classifiers on the subset. Then we combined them into an ensemble (Classifier_C). We compared Classifier_C with our PSRSM_A, and Table 4 shows that the performance of PSRSM_A is quite similar to that of Classifier_C on CB513 dataset, but significantly better on subset with protein length L ∈ [1, 100]. The main difference between the two classifiers was the training set. All training proteins of PSRSM_A were short proteins, they had similar protein lengths, and all lengths belonged to interval [1,100]; conversely, the lengths of Classifier_C training data were randomly distributed.

Table 5 shows the performance of T100 dataset with different lengths based on 6 PSRSMs. 6 protein subsets with different lengths achieved the best performance 79.84%, 84.58%, 87.59%, 87.51%, 83.24%, and 83.93% respectively using their corresponding PSRSM.

Training time analysis.

Another advantage of our method is that the training time was short. Because our training data ASTRAL+CullPDB is a large dataset, it was very slow to train the SVM classifier. We failed to train the SVM classifier on ASTRAL+CullPDB using our server. The computational complexity to train an SVM is

\[ O(SVM) = O(N_s^2 + N_f^2 N + N_s N_f N) \]

Where \( N_s \) is the number of support vectors, \( N_f \) is the feature dimension, and \( N \) is the size of the training set.

After data partitioning and sampling, the number of support vectors \( N_s \), feature dimension \( N_f \), and the size of the training set \( N \) are much smaller. Furthermore, since we trained our base classifiers in parallel, the running time was reduced.

Table 6 shows the training time on each subset of the ASTRAL+CullPDB. \( D_1, D_2, \ldots, D_6 \) were subsets of ASTRAL+CullPDB (Table 7). We failed to train the SVM classifier on the ASTRAL+CullPDB using our server. After data partitioning but before sampling we completed training of SVM classifiers on each subset; more time was required because \( D_6 \) had more amino acids than other subsets. When we used PSRSM, the feature dimension was decreased, and the training time was reduced.

Conclusion and Future Work

In this paper we proposed a novel method, PSRSM, to predict protein secondary structure. The first step of our method was partitioning of the training set into several subsets based on protein length. In the second step, we generated \( k \) ensemble classifiers using the semi-random subspace method. If given a new query protein sequence, our method would select one, and only one, ensemble classifier from \( k \) ensemble classifiers according to length to predict the protein secondary structure. Experimental results showed that the overall performance of PSRSM was better than that of other current state-of-the-art methods. In particular, our method PSRSM is superior to other methods not only in internal regions, but also at boundaries.

Methods

Partitioning the training data. We partitioned the training data into \( k \) different subsets according to the protein sequence length. Let \( X \) denote a protein sequence, and \( L \) denote the length of \( X \). We set \( k \) to 1 partition.

| Methods       | CASP10 | CASP11 | CASP12 | CB513 |
|---------------|--------|--------|--------|--------|
| SPINE-X       | 80.7   | 79.3   | 76.9   | 78.9   |
| PSIPRED       | 81.2   | 80.7   | 78.0   | 79.2   |
| JPRED         | 81.6   | 80.4   | 75.1   | 81.7   |
| DeepCNF       | 84.4   | 84.7   | 82.1   | 82.3   |
| PSRSM         | 85.51  | 85.89  | 85.55  | 84.53  |

Table 1. Q3 accuracy of the tested methods on CASP10, CASP11, CASP12, and CB513 datasets. (The results of SPINE-X, PSIPRED, JPRED, and DeepCNF are taken from the papers.2,27)
points of interval \((0, \infty)\). Let \(r_0 = 0\), \(r_k = \infty\), and \(r_0 < r_1 < \cdots < r_{k-1} < r_k\). These partition points partition interval \((0, \infty)\) into \(k\) intervals without intersection. Let \(R = \{(0, r_0), (r_0, r_1), \cdots, (r_{k-1}, \infty)\} \).
Table 3. PSRSM, DeepCNF, SPIDER3, MUFOLD, PSIPRED and JPRED average Q3 accuracies and Q3 accuracies in the internal regions, and at boundary regions of secondary structures on the T100. The DeepCNF method is available only to proteins with a length of [26, 4000], MUFOLD is [30, 700], and JPRED is [20, 800].

| Method     | Q3(average) | Q3 (internal) | Q3 (boundary) | Website                                      |
|------------|-------------|---------------|---------------|----------------------------------------------|
| DeepCNF    | 82.78       | 85.68         | 73.30         | http://raptorx.uchicago.edu/StructurePropertyPred/predict/ |
| SPIDER3    | 82.41       | 88.25         | 70.72         | http://sparks-lab.org/server/SPIDER3/          |
| MUFOLD     | 84.35       | 89.28         | 74.65         | http://mufold.org/mufold-ss-angle/             |
| PSIPRED    | 76.33       | 82.84         | 63.06         | http://bioinf.cs.ucl.ac.uk/psipred/            |
| JPRED      | 74.45       | 81.42         | 60.25         | http://www.compbio.dundee.ac.uk/jpred4/index.html |
| PSRSM      | 85.09       | 89.89         | 75.33         | http://210.44.144.20:82/protein_PSRSM/default.aspx |

Table 4. Comparison of classifier_C and PSRSM, on CB513.

| Protein length L | PSRSM, Classifier_C | PSRSM, PSRSM | PSRSM, PSRSM | PSRSM, PSRSM | PSRSM, PSRSM |
|------------------|----------------------|--------------|--------------|--------------|--------------|
| [1,100]          | 79.84                | 63.11        | 62.75        | 63.40        | 62.89        | 64.13        |
| (100,200]        | 78.19                | 84.58        | 81.02        | 78.99        | 77.18        | 78.16        |
| (200,300]        | 74.39                | 78.14        | 87.59        | 78.99        | 75.95        | 75.15        |
| (300,400]        | 74.00                | 75.63        | 78.80        | 87.51        | 78.62        | 77.64        |
| (400,500]        | 74.23                | 76.69        | 77.09        | 80.81        | 83.24        | 77.06        |
| L>500            | 73.59                | 75.87        | 75.64        | 76.30        | 77.12        | 83.93        |

Table 5. Q3 accuracy of each ensemble classifier on different proteins with different length in T100 dataset.

| Subset | No sampling | Sampling (PSRSM) |
|--------|-------------|------------------|
| D1     | 7 days      | 1.5 days         |
| D2     | 30 days     | 6 days           |
| D3     | 45 days     | 8 days           |
| D4     | 40 days     | 7 days           |
| D5     | 15 days     | 3 days           |
| D6     | 35 days     | 6.5 days         |

Table 6. Training time on each subset of the ASTRAL + CullPDB.

| Subset | Protein length L | Number of proteins | Number of amino acids |
|--------|------------------|--------------------|-----------------------|
| D1     | [0, 100]         | 2260               | 161952                |
| D2     | (100, 200]       | 5256               | 774167                |
| D3     | (200, 300]       | 3548               | 87753                 |
| D4     | (300, 400]       | 2382               | 822913                |
| D5     | (400, 500]       | 1170               | 519422                |
| D6     | (500, ∞)         | 1058               | 707309                |

Table 7. Subsets of training data ASTRAL + CullPDB.
Let \( D \) denote the training data ASTRAL + CullPDB. Subsets \( D_1, D_2, \ldots, D_k \) and \( D_k \) are defined as follows:

\[
D_i = \{ X | X \in D \land L \in (\tau_{i-1}, \tau_i], \ i = 1, \ldots, k; \tag{4}\n\]

and, \( D_1, D_2, \ldots, D_{k-1} \) and \( D_k \) satisfy \( \bigcup_{i=1}^{k-1} D_i = D \), and \( D_j \cap D_j = \emptyset, \ i \neq j \).

In our experiment, we set \( k = 6 \) and \( R = \{(0, 100], (100, 200], (200, 300], (300, 400], (400, 500], (500, \infty]\) \).

Table 7 shows the number of proteins and amino acids in \( D \) for each protein sequence was represented by a 260-dimensional dataset.

**Training classifiers.** We generated \( t \) random subspaces of \( r \)-dimension, and trained \( t \) SVM base classifiers on each subset \( D \), \( t \) feature subsets are used to train \( t \) base classifiers, and each subset had \( r \) features sampled from the 260-dimensional dataset.

Therefore we got \( k \times t \) SVM base classifiers on \( k \) subsets, we denote these classifiers as a \( k \times t \) matrix, where \( k \) is the number of subsets of the training data.

\[
\begin{bmatrix}
C_{11} & C_{12} & \cdots & C_{1t} \\
C_{21} & C_{22} & \cdots & C_{2t} \\
\vdots & \vdots & \ddots & \vdots \\
C_{k1} & C_{k2} & \cdots & C_{kt}
\end{bmatrix}
\]

(5)

where \( C_{ij} \) is the SVM base classifier trained on the \( j \)th subspace data of subset \( D_i \).

We combined classifiers \( \{C_{ij}\}_{i=1}^{k} \) into a final ensemble classifier by majority vote rule, and thus got \( k \) ensemble classifiers as the final decision on each subset. They are denoted as below:

\[
PSRSM = \begin{bmatrix}
\text{Voting}(C_{11}, C_{12}, \cdots, C_{1t}) \\
\text{Voting}(C_{21}, C_{22}, \cdots, C_{2t}) \\
\vdots \\
\text{Voting}(C_{k1}, C_{k2}, \cdots, C_{kt})
\end{bmatrix}
= \begin{bmatrix}
PSRSM_1 \\
PSRSM_2 \\
\vdots \\
PSRSM_k
\end{bmatrix}
\]

(6)

Here ‘Voting’ means combining classifiers by majority vote rule, \( PSRSM \), represents the final ensemble classifier on subset \( D_i \), and,

\[
PSRSM_i = \text{Voting}(C_{i1}, C_{i2}, \cdots, C_{it}). \tag{7}
\]

In this study The parameters \( t \) is set to 12 base classifiers, and the dimension of subspaces \( r \) is 160 in our experiment.

The publicly available LIBSVM45 software was used to train SVM classifiers. There are several kernel functions, commonly used in SVM: “liner”, “polynomial”, and “radial basis”. In this paper, we used the radial basis function (RBF) as kernel, the form is \( k(x, x_i) = \exp(-\gamma \|x - x_i\|^2) \), where \( \gamma \) is a parameter. \( C \) is another parameter for SVM training; it is the regularization factor that controls the balance between low error and large divided margin.

Parameters \( C \) and \( \gamma \) were decided using the grid search method. The optimal values of the two parameters are 0.9956 and 0.065, respectively.

**Prediction.** Given a new query protein sequence \( X \), and protein sequence length \( L \), our method selected one of \( k \) ensemble classifiers \( \{PSRSM_1, PSRSM_2, \ldots, PSRSM_k\} \) according to the length \( L \) to predict the protein secondary structure of \( X \). Let \( \bar{Y} \) denote the prediction output by \( PSRSM \). Then

\[
\bar{Y} = PSRSM_i(X) \text{ if } L \in (\tau_{i-1}, \tau_i], \ i = 1, 2, \ldots, k, \tag{8}
\]

where, \( PSRSM_i \) is defined as (7).

For example, if a new query protein sequence \( X \) is a short protein and \( L \in (0, \tau_1] \), then the corresponding \( PSRSM_i \), trained on the short protein subset is used to predict its secondary structure. In general, if \( L \in (\tau_{i-1}, \tau_i] \), the \( i \)th classifier \( PSRSM_i \) will be selected from \( k \) ensemble classifiers to predict the protein secondary structure of \( X \).

**Semi-Random Subspace Method (SRSM).** The random subspace method (RSM) is an ensemble construction technique. It was proposed by Ho in 199846. RSM randomly samples a set of low-dimensionality subspaces from the whole original high-dimensional features space, then constructs a classifier on each smaller subspace and finally applies a combination rule for the final decision.

We proposed a semi-random subspace method for protein secondary structure prediction. In our research, each protein sequence was represented by a 260 \( \times \) \( L \) matrix. The \( i \)th column vector represents features of the \( i \)th amino acid residue. We generated \( t \) feature subsets to train \( t \) base classifiers. Each subset had \( r \) features sampled from the 260-dimensional dataset.

Because the original PSSM of the associated residue is an important feature for the base classifier, those 20 dimensions in a central location of 260-dimensional data are fixed for each sampling.

Let \( S \) represent the 260-dimensional features vector, and \( S = (v_1, v_2, \ldots, v_{260}) \). We generated \( t \) subspaces \( \{S_j^k\} \) from \( S \) where \( S_j^k \) represents a feature subset sampled from \( S \), and \( S_j^k = (x_{11}, x_{12}, \ldots, x_{1r}, y_{11}, y_{12}, \ldots, y_{1r}) \), where \( (x_{11}, x_{12}, \ldots, x_{1r}) \) are fixed in each \( S_j \), \( (y_{11}, y_{12}, \ldots, y_{1r}) \) are sampled from \( (v_1, v_2, \ldots, v_{10}) \)
Figure 2. Relationship between Q3 accuracy and dimension of subspace.

and (v141, v142, ..., v260), respectively; here, r and d satisfy r = 2 × d + 20. Let \( L_i = (x_{i1}, x_{i2}, ..., x_{id}) \), \( S_0 = (v_{121}, v_{122}, ..., v_{140}) \), and \( R_i = (y_{i1}, y_{i2}, ..., y_{id}) \), then \( S_i = L_i \cup S_0 \cup R_i \).

Additionally, high diversity of base classifiers can make an ensemble with more accurate decisions; this is because different base classifiers make different errors on different patterns. The diversity of base classifiers is negatively correlated with the similarity of the training data. \( |S_i \cup S_j| \) reflects the similarity of \( S_i \) and \( S_j \) to a certain extent. When \( |S_i \cap S_j| \) is smaller, the similarity between \( S_i \) and \( S_j \) is smaller, and the diversity of base classifiers is higher. Let \( o_i \) be the number of occurrences of \( v_j \) in \( \{S_i\}_{i=1}^{120} \). In our research, it can be proved that, when \( o_i = \frac{td}{120} \) for \( i = 1, 2, ..., 120 \) and \( i = 141, 142, ..., 260 \), the sum \( \sum_{i=1}^{120} \sum_{j=1}^{120} |S_i \cap S_j| \) becomes the minimum. Therefore our method was to generate \( t \) feature subsets randomly, and adjust elements of each subspace to make \( o_i = \frac{td}{120} \) for \( i = 1, 2, ..., 120 \) and \( i = 141, 142, ..., 260 \).

The steps of the proposed semi-random subspace method are as follows.

1. Generating semi-random subspaces

   (1) Let \( L = (v_1, v_2, ..., v_{120}) \), \( S_0 = (v_{121}, v_{122}, ..., v_{140}) \), and \( R = (v_{141}, v_{142}, ..., v_{260}) \) and generate \( d \)-dimensional random subspaces \( \{L_i\}_{i=1}^{t} \) from \( L \), \( \{R_i\}_{i=1}^{t} \) from \( R \), respectively.

   (2) Calculate \( \{o_j\}_{j=1}^{120} \), where \( o_j \) is the number of occurrences of \( v_j \) in \( \{L_i\}_{i=1}^{t} \). Let \( \text{min} = \min \{o_j\}_{j=1}^{120} \) and \( \text{max} = \max \{o_j\}_{j=1}^{120} \).

   (3) Let \( \text{idmin} = \{j | o_j = \text{min} \land j = 1, 2, ..., 120 \} \) and \( \text{idmax} = \{j | o_j = \text{max} \land j = 1, 2, ..., 120 \} \). Then generate a ternary ordered pairs set \( P = \{ (i, j, k) | v_j \notin L_i \land v_k \notin L_i \land j \in \text{idmin} \land k \in \text{idmax} \} \).

   (4) Randomly select a ternary ordered pair \( (i, j, k) \) from \( P \), insert feature \( v_j \) to \( L_i \), and delete \( v_k \) from \( L_i \). Then return to step (1) until \( o_i = o_2 = \cdots = o_{120} \).

   (5) Repeat (2), (3) and (4) on \( \{R_i\}_{i=1}^{t} \) and \( R \).

   (6) \( S_i = L_i \cup S_0 \cup R_i \), \( i = 1, 2, ..., t \).

2. Construct \( t \) classifier \( \{C_i\}_{i=1}^{t} \) from the corresponding \( t \) random subspaces.

3. Combine classifiers \( \{C_i\}_{i=1}^{t} \) by majority vote rule.

There are two parameters to be determined for the semi-random subspace method, i.e., the number of subspaces \( t \), and dimension of subspaces \( r \).

Since \( D_t \) was smaller than other subspaces, the training time on \( D_t \) was shorter than on other subspaces. Therefore we conducted a series of experiments on \( D_t \) to determine \( t \) and \( r \). We fixed \( t = 12 \), because it requires \( t^d \) to be divided by 120, it is easy to set \( t \) or \( r \). Experimental results on the CB513 dataset showed that with increasing \( r \) the Q3 accuracy increased, but when \( r > 160 \), the Q3 accuracy increased slowly (Fig. 2) and the training time must be much longer. So we determine \( r = 160 \) as the dimension of subspaces in our experiment.

Input features. The PSSM of a protein sequence represents homolog information affiliated with its aligned sequences. We used the PSI-BLAST program to generate the PSSM data. PSI-BLAST used BLOSUM62 evolutionary matrix to search a reduced version of the NCBI’s non-redundant (NR) database filtered at 90% sequence similarity, in order to find the variability of the residue within a multiple sequence alignment. PSI-BLAST parameters was set with threshold \( h = 0.001 \) and \( j = 3 \) iterations. The resulting PSSMs were a \( 20 \times L \) matrix, where \( L \) is the protein length and 20 is the number of amino acid types.

A sliding window of consecutive amino acids was used to obtain residue sequence information and predict the secondary structure of the central residue. Each residue was encoded by a vector of dimension \( 20 \times w \), where \( w \) is the sliding window size and is an odd number. The window was shifted from residue to residue through the protein chain. In this paper, the sliding window length \( w \) was set to 13. To use the first and last six amino acids,
we inserted six zeros before and behind each protein sequence. Therefore each protein sequence was represented by a $260 \times I$ matrix, and the $i$th column vector represented the protein features associated with the $i$th residue. Secondary structure assignment was done with the DSSP. DSSP program defines eight states for secondary structure ($H$, $E$, $B$, $T$, $S$, $L$, $G$, and $I$) that are reduced to three states ($H$, $E$, and $C$) by different predictive methods. We used the following reductions: $H$, $G$ and $I$ to helix ($H$); $E$ and $B$ to beta strands ($E$); all the rest to coil ($C$).

**Availability.** http://210.44.144.20:82/protein_PSRSM/default.aspx.

**References**

1. Alberts B. et al. Molecular biology of the cell, 5th ed. New York: Garland Science (2008).
2. Yang, Y. et al. Sixty-five years of the long march in protein secondary structure prediction: the final stretch? Briefings in Bioinformatics (2016).
3. Kabsch, W. & Sander, C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22, 2577–2637 (1983).
4. Fasman, G. D. & Chou, P. Y. Prediction of protein conformation: consequences and aspirations. Biochemistry 13, 222–245 (1974).
5. Chou, P. Y. & Fasman, G. D. Conformational parameters for amino acids in helical, beta-sheet, and random coil regions calculated from proteins. Biochemistry 13, 211–222 (1974).
6. Garnier, J., Gibrat, J. F. & Robson, B. GOR method for predicting protein secondary structure from amino acid sequence. Methods in Enzymology 266, 540–553 (1996).
7. Jones, D. T. Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol. 292, 195–202 (1999).
8. Altschul, S. F. et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 25, 3389–3402 (1997).
9. Yoo, P. D., Zhou, B. B. & Zomaya, A. Y. Machine learning techniques for protein secondary structure prediction: an overview and evaluation. Current Bioinformatics 3, 74–86 (2008).
10. Holley, L. H. & Karplus, M. Protein secondary structure prediction with a neural network, Proc. Natl. Acad. Sci. USA 86, 152–156 (1989).
11. Qian, N. & Sejnowski, T. J. Predicting the secondary structure of globular proteins using neural network models. J. Mol. Biol. 202, 865–884 (1988).
12. Kneller, D., Cohen, F. & Langridge, R. Improvements in protein secondary structure prediction by an enhanced neural network. J. Mol. Biol. 214, 171–182 (1990).
13. Malekpour, S. A., Naghizadeh, S., Pezeshki, H., Sadeghi, M. & Elslahihi, C. Protein secondary structure prediction using three neural networks and a segmental semi markov model. Mathematical Biosciences 217, 145–150 (2009).
14. Wu, Q., Sui, H., Yang, B. & Qian, W. Improving protein secondary structure prediction using a multi-modal bp method. Computers in Biology & Medicine 41, 946–959 (2011).
15. Asai, K., Hayamizu, S. & Handa, K. Prediction of protein secondary structure by the hidden markov model. Computer Applications in the Biosciences Cabios 9, 141–146 (1993).
16. Won, K. J. et al. An evolutionary method for learning HMM structure: prediction of protein secondary structure. BMC Bioinformatics 8, 1–13 (2007).
17. Aydin, Z., Altunbasak, Y. & Borodovsky, M. Protein secondary structure prediction for a single-sequence using hidden semi-Markov models. BMC Bioinformatics 7, 178 (2006).
18. Kim, H. & Park, H. Protein secondary structure prediction based on an improved support vector machines approach. Protein Eng. 16, 553–560 (2003).
19. Ward, J. J., McGuffin, L. J., Buxton, B. F. & Jones, D. T. Secondary structure prediction with support vector machines. Bioinformatics 19, 1650–1655 (2003).
20. Guo, J., Chen, H., Sun, Z. & Lin, Y. A novel method for protein secondary structure prediction using dual - layer SVM and profiles. Protein: Struct. Funct. Bioinform. 7, 738–743 (2004).
21. Hua, S. & Sun, Z. A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. J. Mol. Biol. 308, 397–407 (2001).
22. Tan, Y. T. & Rosdi, B. A. Fpga-based hardware accelerator for the prediction of protein secondary class via fuzzy k-nearest neighbors with lempel-ziv complexity based distance measure. Neurocomputing 148, 409–419 (2015).
23. Bouziane, H., Messabih, K. & Chouaaria, A. Profiles and majority voting-based ensemble method for protein secondary structure prediction. Evolutionary Bioinformatics 7, 171–188 (2011).
24. Zhou, J. & Troyanskaya, O. D. Supervised and Convolutional Generative Stochastic Network for Protein Secondary Structure Prediction. Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21–26 June 2014. JMLR Proceedings 32, 745–753 (2014).
25. Spencer, M., Eickholt, J. B. & Cheng, J. A Deep Learning Network Approach to ab initio Protein Secondary Structure Prediction. IEEE/ACM Trans. Comput. Biol. Bioinform. 12, 103–112 (2015).
26. Lee, H., Grosse, R., Ranganath, R. & Ng, A. Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14–18 (2009).
27. Wang, S. et al. Protein secondary structure prediction using deep convolutional neural fields. Scientific Reports, https://doi.org/10.1038/srep18892 (2016).
28. Wang, S., Li, W., Liu, S. & Xu, J. RaptorX-property: a web server for protein structure property prediction. Nucleic Acids Research 44, W430–W435, https://doi.org/10.1093/nar/gkw306 (2016).
29. Fang, C., Shang, Y. & Xu, D. MUFOLD-SS: New deep inception-inside-inception networks for protein secondary structure prediction. Proteins 86, 592–598 (2018).
30. Heffernan, R., Yang, Y., Paliwal, K. & Zhou, Y. Capturing Non-Local Interactions by Long Short Term Memory Bidirectional Recurrent Neural Networks for Improving Prediction of Protein Secondary Structure, Backbone Angles, Contact Numbers, and Solvent Accessibility. Bioinformatics 33, 2842–2849 (2017).
31. Fox, N. K. SCOPE: Structural classification of proteins-extended, integrating SCOP and ASTRAL data and classification of new structures. Nucleic Acids Research 42, 304–309 (2014).
32. Wang, G. & R. D. Jr. PISCES: recent improvements to a PDB sequence culling server. Nucleic Acids Research, 33(Web Server issue), W94–W98 (2005).
33. Moult, J., Fidelis, K., Kryshtafovych, A. & Tramontano, A. Critical assessment of methods of protein structure prediction (CASP)-round X. Proteins: Structure, Function, and Bioinformatics 79, 1–5 (2012).
34. Moult, J., Fidelis, K., Kryshtafovych, A. & Tramontano, A. Critical assessment of methods of protein structure prediction (CASP)-round XI. Proteins: Structure, Function, and Bioinformatics 82, 1–6 (2014).
35. Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T. & Tramontano, A. Critical assessment of methods of protein structure prediction (CASP): progress and new directions in Round XII. Proteins: Structure, Function, and Bioinformatics 84(S1), 4–14 (2016).
36. Cuff, J. A. & Barton, G. J. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. Proteins: Structure, Function, and Bioinformatics 34, 508–519 (1999).
37. Kedarisetti, K. D., Kurgan, L. & Dick, S. Classifier ensembles for protein structural class prediction with varying homology. Biochem. Biophys. Res. Comms. 348, 981–988 (2006).
38. Faraggi, E., Zhang, T., Yang, Y., Kurgan, L. & Zhou, Y. SPINE X: improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles. J. Comp. Chem. 33, 259–267 (2012).
39. Drozdetskiy, A., Cole, C., Procter, J. & Barton, G. J. JPred4: a protein secondary structure prediction server. Nucleic Acids Res. gkv332 (2015).
40. McGuffin, L. J., Bryson, K. & Jones, D. T. The PSIPRED protein structure prediction server. Bioinformatics 16, 404–405 (2000).
41. Meyer, O., Bischl, B., & Weihs, C. Support Vector Machines on Large Data Sets: Simple Parallel Approaches. Data Analysis, Machine Learning and Knowledge Discovery. Springer International Publishing, 87–95 (2014).
42. Li, W., Kimch, I. N., Karplus, P. A. & Grishin, N. V. Chseq: a database of chameleon sequences. Protein Science 24, 1075–1086 (2015).
43. Vapnik, V. N., Statistical learning theory. Encyclopedia of the Sciences of Learning (2008).
44. Chang, C. & Lin, C. LIBSVM: A library for support vector machines. ACM. 2, 1–27 (2011).
45. Ho, T. K. The Random Subspace Method for Constructing Decision Forests. IEEE Transactions on Pattern Analysis & Machine Intelligence 20, 832–844 (1998).

Acknowledgements
The work is supported by the National Natural Science Foundation of China (61375013), Shandong Provincial Natural Science Foundation, China (ZR2013FM020).

Author Contributions
Y.M. designed, implemented the algorithm, and wrote the manuscript. Y.L. supervised the whole experiments, collected all the experimental data, and performed part of experiments. J.C. performed part of experiments, and designed our web site. All authors read and approved the final manuscript.

Additional Information
Supplementary information accompanies this paper at https://doi.org/10.1038/s41598-018-28084-8.

Competing Interests: The authors declare no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2018