Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: 31/01/2021
Scientific article

OXFORD

Laboratory of Bioinformatics II

Comparison of probabilistic and non-linear methods for the prediction of protein secondary structure from sequence

Aigerim Rymbekova

Department of Pharmacy and Biotechnology, University of Bologna, Bologna, 40126, Italy

Abstract

Motivation: Protein secondary structure prediction is one of the most important and challenging problems in bioinformatics. A plenty of different techniques have been applied to tackle the problem and some have gained substantial success in the research area. In this paper we present the comparison and the analysis of prediction performances between GOR method, a probabilistic statistical analysis, and Support Vector Machines method, a machine learning based method.
Results: The overall percentage of residues for three-state secondary structures to be predicted correctly was slightly higher for the SVM method 70.3% versus 68.2% for the GOR method.
Contact: aigerim.rymbekova@studio.unibo.it
Supplementary information: Supplementary data are available at *https://github.com/Rymbekova/lb2-2020-project-Rymbekova*

1 Introduction

Proteins play a key role in almost all biological processes and mediate a big number of functions inside the cell. Protein structure is essential for the understanding of protein function. There are 20 different amino acids that form proteins in nature (Alberts, 2008). 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 comprising the first level of organization. Protein secondary structure refers to the local conformation proteins' polypeptide backbone. Sander developed a secondary structure assignment method Dictionary of Secondary Structure of Proteins (DSSP) (Kabsch and Sander, 1983), which automatically assigns secondary structure into eight states according to hydrogen-bonding patterns. These eight states are often further simplified into three states: two regular secondary structure states, alpha-helix and beta-strand, and one irregular secondary structure type, the coil region. Protein secondary structure prediction is usually evaluated by Q3 accuracy (Im, 2008), which measures the percentage of residues for three-state secondary structures to determine whether they have been predicted correctly, one of the metrics we will use for the evaluation of our predictors in this project. Protein secondary structure prediction provides a significant first step toward tertiary structure prediction, as well as offering information about protein activity, relationships, and functions. Accurately and reliably predicting structures from protein sequences is one of the most challenging tasks in computational biology.

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 (Yang, 2016). Many statistical approaches and machine learning approaches have been developed to predict secondary structure. The GOR method (Garnier *et al.*, 1996) formalizes the secondary structure prediction problem within an information-theoretic framework. Position specific scoring matrix (PSSM) (Gribskov *et al.*, 1987) based on PsiBLAST (Altschul, 1997) reflects evolutionary information and has made the most significant improvements in protein secondary structure prediction. Machine learning methods have been developed to predict protein secondary structure, and exhibit good performance by exploiting evolutionary information and statistic information about amino acid sub-sequences (Yoo *et al.*, 2008). For example, many neural network methods (Holley and Karplus, 1989), hidden Markov model (Asai *et al.*, 1993), Support Vector Machines (SVM) (Cortes and Vapnik, 1995) and K-nearest neighbors (Tan and Rosdi, 2015) have had substantial success, and Q3 accuracy has reached to 80%.

Here, in this project we will focus and analyse in detail two of methods mentioned above: GOR method, a probabilistic statistical analysis, and SVM method, a machine learning based method. Using them we will endow a set of proteins with predicted secondary structure, presenting the identical data to both models. Then we will compare the results with the DSSP assignments of secondary structure. Finally, we will evaluate their performance based on several metrics, also training and prediction computational time.

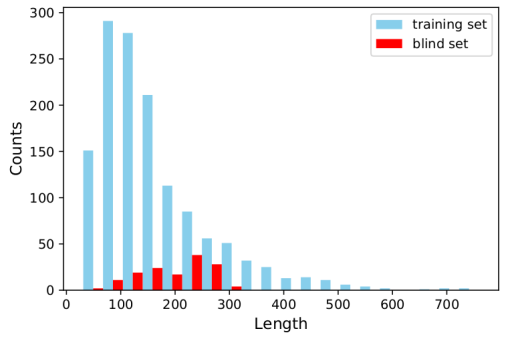

Fig. 1. The comparison of sequence length between training and test sets.

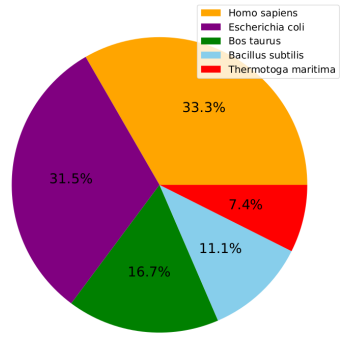

Fig. 2. The taxonomic classification of the test set sequences.

2 Methods and Materials

2.1 Training set

In order to train our models we used the publicly available Jpred4 training set (Drozdetskiy *et al.*, 2015). It consists of 1348 fasta sequences and of the corresponding secondary structure assignments obtained with DSSP v. 3.0.0 (Kabsch and Sander, 1983) from structures deposited in the Protein Data Bank (PDB) (Berman *et al.*, 2000). DSSP secondary structure classes are mapped to three categories in this dataset: helix as H, strand as E, and coil as C (Cuff and Barton, 2000).

2.2 Test set

To create a test set using the advanced search on PDB we downloaded all structures that meet the following criteria:

- release date after January 1, 2015
- experimental method: X-ray diffraction
- resolution better or equal to 2.5 Å
- chain length between 50 and 300 residues included
- polymer entity type: protein.

Firstly, we downloaded the PDB IDs list and retrieved the correnspondent fasta sequences. Secondly, we extracted all chain sequences in fasta format and discarded chains that are identical or do not satisfy the length constraint or contain unidentified residues as an X character. Then we used Blastclust (Dondoshansky and Wolf, 2002) on downloaded fasta sequences for reducing internal redundancy to 30% sequence identity and 50% coverage. Blastclust begins with pairwise matches and then places a sequence in a cluster if the sequence matches at least one sequence already in the cluster. The cluster representatives were aligned to the training set using BlastP v. 2.5.0 (Altschul *et al.*, 1990) with E-value threshold of 0.01 and the proteins with more than 30% sequence identity have been filtered out. Finally, from the chains survived all filtering steps random 150 protein chains were selected for the blind set with corresponding fasta sequences.From selected and downloaded PDB structures we generated DSSP secondary structure assignment files using mkdssp v. 2.2.1 (Kabsch and Sander, 1983) with the same mapping strategy as shown by Jpred4 authors. The preliminary statistical analysis of the blind set was performed (Figures 1, 2, 3).

Generation of sequence profiles

The sequence profiles were generated for all the sequences in the training and test sets with PsiBlast v. 2.5.0 against the entire UniProtKB/SwissProt database (release 2020_06) (UniProt, 2019). A sequence profile is a compact representation of a protein family specifically built for a target protein sequence. It is produced converting multiple sequence alignments

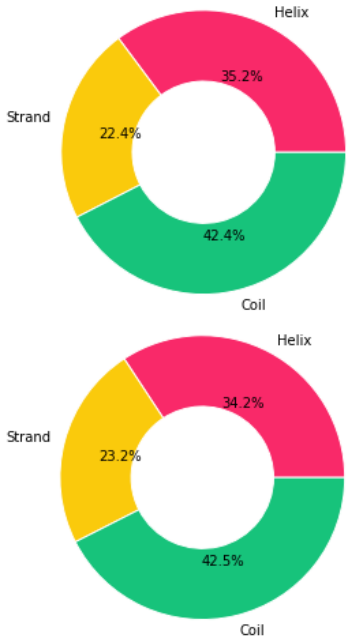

Fig. 3. The overall proportions of residues in helix, strand, and coil conformations in the training (above) and test sets.

into PSSMs. Each of the 20 amino acids in each position of the alignment is scored according to the frequency in which it occurs in the original alignment (Gribskov *et al.*, 1987). The output was obtained from the multiple sequence alignment of sequences detected above score threshold of 0.01 for three iterations. Once we obtained a checkpoint/PSSM file for each protein we extracted the sequence profile normalizing each value by dividing for 100. For some sequences lacking enough homologs PsiBlast failed to create a profile resulting in rows and columns values equal to 0, these profiles were discarded. In the end, 1251 profiles were in the training set and 144 profiles were in the blind set.

2.3 The GOR method

The GOR (Garnier-Osguthorpe-Robson) method is one of the first major methods proposed for protein secondary structure prediction from sequence (Garnier *et al.*, 1996). The algorithm was modified and continuously improved for 30 years including the triplet statistics within a window. Another important advance was including multiple sequence alignments from PsiBlast program that allowed to take into consideration evolutionary information for the secondary structure prediction.

The GOR method is an information theory model for secondary structure prediction, and is based on the statistical propensities of residues for secondary structure conformations. The information function I calculates the probability of a residue R to be in the conformation S, such that:

$$I(S;R) = \log \frac{P(S|R)}{P(S)} \tag{1}$$

where $P(S|R)$ is the conditional probability of observing residue R (where R is one of 20 possible amino acids) in conformation S (where S is one of three secondary structure classes: H as helix, E as extended or C as coil), and $P(S)$ is the probability of conformation S. Using the chain rule we get :

$$I(S;R) = \log \frac{P(R,S)}{P(S)P(R)} \tag{2}$$

We should also take into consideration the context of each residue: how much its neighbors influence the structural conformation of a given R:

$$I(S, R_{-d}, ..., R_d) = \log \frac{P(S|R_{-d}, ..., R_d)}{P(S)P(R_{-d}, ..., R_d)} \tag{3}$$

where d is the number of residues taken into consideration immediately before and after the central residue (8 nearest neighboring residues on each side, or a 17 residues long sliding window). Computing all the joint probabilities of S and all the possible residues in all the positions of the window is computationally expensive as it would be an exponential number of possible configurations and a very large sequence database to estimate reliable distributions. To overcome this problem, we can assume statistical independence of all residues within the window, so that:

$$I(S; R_{-d}, \ldots, R_d) \approx \sum_{k=-d}^{d} I(S; R_k) \tag{4}$$

Meaning, the window-based information function is equal to the sum of the single residue functions. The parameters needed to compute the equation above are:

- $P(R_k, S)$: probability of observing a residue of type R at position k in the window and the central residue R_0 in conformation S:
- $P(R_k)$: probability of observing a residue of type R at position k in the window;
- $P(S)$: probability of observing conformation S.

Training

For each position along a given sequence profile, 17 profile positions around it will be considered, defining a sliding window. For positions proximal to the extremities of the input sequence, the rows of the sliding window are filled with zeros. So that we will have matrices associated with secondary structure conformation of each residue being the centre of the window. At the beginning of the training, the residue counts matrix will be filled with pseudocounts (value 0.01) to avoid divisions by zero later. This matrix will represent the secondary structure conformation in the first axis, the position along the sliding window in the second axis, and the residue type in the third axis. After processing the training set, the counts will be normalized to frequencies. The values of frequencies are the joint probabilities of observing a residue of a certain type in a certain position of the sliding window when the central residue of the window is in a certain conformation. The frequencies will be then transformed in log-propensities - the joint probabilities of frequencies divided by the product of the independent probabilities of observing a certain residue in a certain position of the window and the overall probability of observing a certain conformation. The value is positive if the joint probability is higher than expected and negative otherwise. The trained model consists of the matrix of log-propensities.

Prediction

Having a sequence profile and a matrix for each position along the sequence using a sliding window, we can try to predict the secondary structure of a given sequence. During the prediction phase, each residue position of all the query sequences is analyzed, associating it with the conformation S characterized by the highest value in the information matrix:

$$S = \arg\max_S \sum_{k=-d}^{d} I(S; R_k) \tag{5}$$

2.4 Support Vector Machines method

The SVMs are mathematical objects that can solve non-linearly separable problems in a way that they can find a hyperplane (HP) separating two classes that satisfies the conditions of maximum margin (two parallel hyperplanes allowing the maximum distance between the decision boundary and the nearest class points in the space). A HP can be defined as the set of points $\vec{x}$ that have a fixed projection on a given vector $\vec{w}$ perpendicular to it:

$$\vec{w}\vec{x} + b = 0 \tag{6}$$

Given an HP, we are able to classify a new point x as a member of a class on the basis of computing a scalar product wx + b. If the scalar product equals 1, it is the positive class, if negative, the point belongs to the negative class. If the scalar product equals 0, then the point lies on the HP. The problem that we have here is maximizing the margin, so that the norm of w is as low as possible while leaving the points of two classes on opposite sides of the decision boundary:

$$y_i(\vec{w}\vec{x} + b) \geqslant 1 \forall i \tag{7}$$

where x is a point and y_i is its class (-1,1).

Minimizing the norm of w under the condition (7) is a constrained optimization problem, so we can use the the Lagrange multipliers technique under the Karush-Kuhn-Tucker (KKT) conditions to solve it (Karush, 1939; Kuhn and Tucker, 1951). The minimization of the Lagrangian function can be solved by instead maximizing the dual Lagrangian problem:

$$\begin{aligned} &\bar{L}(\alpha_i) = -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y^i y^j x^i x^j + \sum_i \alpha_i \\ &\alpha_i[1 - y^j(\langle w, x^i\rangle + b)] = 0 \; \forall \, i \\ &\alpha_i \geq 0 \; \forall \, i \end{aligned} \tag{8}$$

While training we can use a soft margin classification, where a certain degree of margin violation is allowed and slack variables are introduced. The previous hard margin notation now can be modified to incorporate slack variables:

$$\begin{aligned} &\frac{1}{2}\|w\|^2 + C\sum_i \xi_i \\ &y^i[\langle w, x^i\rangle + b] \geq 1 - \xi_i \quad \xi_i \geq 0, \forall i \end{aligned} \tag{9}$$

where C is a hyperparameter that controls the tradeoff between margin width and margin violation. The dual Lagrangian for soft margin SVM includes an additional upper bound on the value of the α_i that must be less or equal C.

If the training set data is not linearly separable, we can take advantage of the kernel trick: a non-linearly separable data remapped in a space ϕ with higher dimensionality than the original feature space where it can be linearly separated. The kernel function $K(\vec{x}, \vec{y})$ is the scalar product $\phi(\vec{x}, \vec{y})$ in the new space. The most commonly used kernel functions are linear, polynomial and radial basis function (RBF). In this project, the implemented kernel was the RBF kernel with different combinations of the γ and C hyperparameters:

$$K(\vec{x}, \vec{y}) = e^{(-\gamma \|\vec{x} - \vec{y}\|^2)} \tag{10}$$

Implementation

Generally, SVMs were not designed to perform multi-class classification analysis like predicting protein secondary structure. This can be overcame by one-vs-all strategy that consists in fitting one classifier per class. For each classifier, the class is fitted against all the other classes. In addition to its computational efficiency, this is the most commonly used strategy for multi-class classification.

To find a combination of SVM hyperparameters that results in the best generalization performance, a grid search was performed with the various combinations of the γ and C hyperparameters. The selection of the best combination was based on the different metrics for three secondary structure classes.

In the training phase the input pre-processing steps were similar to the ones shown in the GOR method training. The residue counts matrices were converted to vectors by concatenating each row. This resulted in a feature vector of 340 elements, frequencies between 0 and 1 for each residue in the training set associated with a secondary structure class. The libsvm (Chang and Jen Lin, 2011), a software for support vector classification (including multi-class classification) via svm-train module takes in input a training file and produces a SVM model file as an output. The svm-predict module instead takes a testing file and the trained model to give us a prediction file in output. For the training, we provided these parameters: SVM type: multi-class classification, kernel type: RBF, gamma and C: 0.5/2 and 2/4. For each parameter setting, libsvm provided cross-validation accuracy and the parameters with the highest results were returned. The best parameters then will be used for generating the final model and predicting the blind set sequences.

2.5 k-fold Cross-validation

Cross-validation is a re-sampling procedure applied to evaluate machine learning models on a limited data sample. The goal of cross-validation is to test the model's ability to predict new data that have not been used during the training, in order to avoid overfitting. It gives an idea on whether the produced model will be general enough to correctly predict the blind set. The idea is to randomly divide the training set into k groups of approximately equal size. For our purpose, the training set was split into 5 groups, so that each observation was assigned to an individual group and stayed in that group for the duration of the procedure. This means that each sample is given the opportunity to be used as a test set 1 time and used to train the model k-1 times. The results of a k-fold cross-validation run will be summarized as the mean of the model skill scores including the standard deviation.

2.6 Performance evaluation

The performance of the predictors will be evaluated using following metrics: accuracy, Matthews correlation coefficient (MCC) (Matthews, 1975) and Q3 score, the multi-class accuracy as the fraction of amino acids for which the secondary structure conformation was correctly predicted (Im, 2008). The MCC is a more reliable statistical rate which accounts for all of the four categories: true positives (TP), false negatives (FN), true negatives (TN) and false positives (FP). When the MCC returns the value of 1, a classifier makes perfect predictions, 0 no better than random prediction and -1 indicates total disagreement between prediction and observation. This rate is not affected by class imbalance like the accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

3 Results

3.1 The GOR method

The GOR predictor generally performed better on the test set. The 5-fold cross-validation on the training set showed accuracy of 0.64 for helix, 0.52 for strand and 0.72 for coil (Table 1). The performance on the blind set was better for all classes: 0.77 for helix, 0.69 for strand and 0.79 for coil (Table 2). The overall accuracy and MCC performance in both cases were better on helix and coil than on strand. The MCC performance on the strand conformation was negligibly better on the training set than on the blind set. Q_3 score was 53.6% on the training set and 68.2% on the blind set.

Table 1. The GOR method performance on the training set

	Multi-class	Helix	Strand	Coil
Accuracy	0.633 ± 0.01	0.649 ± 0.01	0.524 ± 0.01	0.727 ± 0.02
MCC	0.485 ± 0.05	0.544 ± 0.03	0.445 ± 0.03	0.467 ± 0.02

Table 2. The GOR method performance on the blind set

	Multi-class	Helix	Strand	Coil
Accuracy	0.751	0.771	0.691	0.793
MCC	0.535	0.595	0.441	0.571

3.2 The SVM method

The SVM method outperformed the GOR method in predicting on average all classes except the coil conformation. The 5-fold cross-validation on the training set with the best-performing hyperparameters combination (γ = 0.5 and C = 2) showed accuracy of 0.71 for helix, 0.71 for strand and 0.66 for coil (Table 3). The performance on the blind set was better for all classes: 0.79 for helix, 0.79 for strand and 0.71 for coil (Table 4). The overall accuracy and MCC performance in both cases were better on helix and strand than on coil. Q_3 score was 65.0% on the training set and 70.3% on the blind set.

Table 3. The SVM method performance on the training set

	Multi-class	Helix	Strand	Coil
Accuracy	0.699 ± 0.03	0.719 ± 0.01	0.719 ± 0.01	0.661 ± 0.01
MCC	0.568 ± 0.02	0.622 ± 0.01	0.557 ± 0.01	0.526 ± 0.02

Table 4. The SVM method performance on the blind set

	Multi-class	Helix	Strand	Coil
Accuracy	0.768	0.795	0.794	0.716
MCC	0.597	0.665	0.590	0.536

4 Discussion

The GOR method performance was better for all classes on the test set according to accuracy and the MCC values. Generally, it seems like it was more difficult to predict strand for the model. In both cases, the performance was slightly higher for helix and coil. One of the possible explanations - long-range amino acid interactions, which may overwrite local sequence propensity of secondary structures (Minor and Kim, 1996; Munoz *et al.*, 1996). The most common current methods (including the ones using in this project) assign a secondary structure to a window of a local segment and thus usually do not explicitly consider long-range interactions of amino acids (Kihara, 2005). Therefore, β-strands prediction is influenced the most as the formation of them is affected by long-range interactions: the hydrogen bonds are formed between residues that have high contact order, they are separated by many residues along a chain so these contacts are outside the sliding window (Rashid *et al.*, 2016).

In case of the SVM, the performance was also higher for all classes on the test set according to all metrics. The overall percentage of residues for three-state secondary structures that have been predicted correctly was slightly higher for the SVM method: 70.3% versus 68.2%. The model outperformed the GOR method in predicting on average all classes except the coil conformation in both training and test sets. It has been shown that the prediction of the coil conformation is characterized by the highest average error rates (Zhang *et al.*, 2011). The prediction quality is also affected by the position with respect to the protein surface and the flexibility of residues. The solvent exposed and flexible coils are predicted with a better accuracy, but may also be misclassified as strand. This trend is common to machine learning-based secondary structure prediction algorithms (Rashid *et al.*, 2016; Zhang *et al.*, 2011).

To conclude, our findings are in agreement with previous studies reporting that the predictions for the helix residues are more accurate with lower average error rate than the rest (Rashid *et al.*, 2016; Zhang *et al.*, 2011; Rademaker, 2020).

5 Conclusion

Protein secondary structure prediction indeed has been an area of intense research interest. However, despite advances in recent methods conducted on large datasets, the estimated upper limit accuracy is yet to be reached. In our work, both methods proved to be relatively good predictors, showing $Q3$ score of 68.2% and 70.3%. The higher performance of SVM may be linked to an increase of the time complexity. The GOR method is sufficiently fast to produce results: the training phase was a matter of minutes similar to the cross-validation step and the prediction phase. The SVM training is much more computationally expensive: several days while the prediction took about a few minutes.

References

Alberts, B. (2008). Molecular biology of the cell. *New York: Garland Science*.

Altschul, S. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research*, **25**, 3389–3402.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, **215**, 403–410.

Asai, K., Hayamizu, S., and Handa, K. (1993). Prediction of protein secondary structure by the hidden markov model. *Computer Applications in the Biosciences Cabios*, **9**, 141–146.

Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., I.N., S., and Bourne, P. (2000). The protein data bank. *Nucleic Acids Research*, **28**, 235–242.

Chang, C. and Jen Lin, C. (2011). Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, **2**, 1–27.

Cortes, C. and Vapnik, V. (1995). Support vector networks. *Machine Learning*, **20**, 273–297.

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

Dondoshansky, I. and Wolf, Y. (2002). Blastclust (ncbi software development toolkit). *NCBI: Bathesda Md.*

Drozdetskiy, A., Cole, C., Procter, J., and Barton, G. (2015). Jpred4: a protein secondary structure prediction server. *Nucleic Acids Research*, **43**, W389–W394.

Garnier, J., Gibrat, J. F., and Robson, B. (1996). Gor method for predicting protein secondary structure from amino acid sequence. *Methods Enzymol.*, **266**, 540–553.

Gribskov, M., McLachlan, A. D., and Eisenberg, D. (1987). Profile analysis: detection of distantly related proteins. *Proceedings of the National Academy of Sciences*, **84**, 4355–4358.

Holley, L. H. and Karplus, M. (1989). Protein secondary structure prediction with a neural network. *Proc. Natl. Acad. Sci. USA*, **86**, 152–156.

Im, I. G. (2008). *Predicting Protein Secondary Structure Using Markov Chain Monte-Carlo Simulation.* ProQuest.

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

Karush, W. (1939). Minima of functions of several variables with inequalities as side constraints. *(M.Sc. thesis). Dept. of Mathematics, Univ. of Chicago, Chicago, Illinois.*

Kihara, D. (2005). The effect of long-range interactions on the secondary structure formation of proteins. *Protein Sci.*, **14**, 1955–1963.

Kuhn, H. and Tucker, A. W. (1951). Nonlinear programming. *Proceedings of 2nd Berkeley Symposium. Berkeley: University of California Press*, page 481–492.

Matthews, B. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta*, **405**, 442–451.

Minor, J. D. and Kim, P. (1996). Context-dependent secondary structure formation of a designed protein sequence. *Nature*, **380**, 730–734.

Munoz, V., Cronet, P., Lopez-Hernandez, E., and Serrano, L. (1996). Analysis of the effect of local interactions on protein stability. *Fold. Des.*, **1**, 167–178.

Rademaker, D. (2020). The future of protein secondary structure prediction was invented by oleg ptitsyn. *Biomolecules*, **10**, 910.

Rashid, S., Saraswathi, S., Kloczkowski, A., Sundaram, S., and Kolinski, A. (2016). Protein secondary structure prediction using a small training set (compact model) combined with a complex-valued neural network approach. *BMC Bioinformatics*, **17**, 362.

Tan, Y. T. and Rosdi, B. A. (2015). 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.

UniProt, C. (2019). Uniprot: a worldwide hub of protein knowledge. *Nucleic Acids Research*, **47**, 506–515.

Yang, Y. (2016). Sixty-five years of the long march in protein secondary structure prediction: the final stretch? *Briefings in Bioinformatics*.

Yoo, P. D., Zhou, B. B., and Zomaya, A. Y. (2008). Machine learning techniques for protein secondary structure prediction: an overview and evaluation. *Current Bioinformatics*, **3**, 74–86.

Zhang, H., Zhang, T., Chen, K., and Kedarisetti, K. (2011). Critical assessment of high-throughput standalone methods for secondary structure prediction. *Brief. Bioinform.*, **12**, 672–88.