Bioinformatics, 2022, 1–8
doi: 10.1093/bioinformatics/xxxxx
Advance Access Publication Date: : 02/02/2022
Scientific article

Laboratory of Bioinformatics II

Testing the Garnier-Osguthorpe-Robson method in comparison to Support Vector Machines for protein secondary structure prediction from primary sequence

Amrou Abas[1,*]

[1]Department of Pharmacy and Biotechnology, University of Bologna, Bologna, Italy.

Address: Via San Donato 15, Bologna, Italy.

*To whom correspondence should be addressed.

Received on 02.02.2022

Abstract

Motivation: Nowadays, providing a correct protein secondary structure prediction is one of the most challenging scopes in bioinformatics, and while the invention of large-scale sequencing techniques paved the way to a massive increase in protein sequence data. The experimental aspect (using X-ray crystallography and NMR for the prediction of the 3D structure for example) is still costly and quite time-consuming, in addition to the fact that there is a large gap between the increasing number of discovered sequences and resolved structures and that cannot be reduced using solely means of experimental approaches. Therefore, two different methods for the secondary structure prediction were tested: the Garnier-Osguthorpe-Robson (GOR) and the Support Vector Machine (SVM) methods which are based on Bayesian statistics and machine learning respectively.
Results: The outcome exhibits how the machine-learning-based method carries out a slightly better performance than the GOR method, giving more accurate predictions with respect the blind test set.

Contact: amrou.abas@studio.unibo.it
Supplementary information: Supplementary data are available clicking here.

1 Introduction and approach

Proteins are macromolcules which are made up of linear chains of amino acids. These biopolymers perform various vital functions such as catalysing metabolic reactions, DNA replication, responding to stimuli, providing structure to cells and organisms, and transporting molecules from one location to another.

There are four levels of organization of these macromolecules, the first one namely is the primary structure, which is a sequence of amino acids in a polypeptide chain. This organization level by itself does not offer very valuable intel about shape of the protein, the surface area that can be accessible by a solvent or even the surfaces' characteristics[1]. To top this off, the loss of the higher order organizations leads to render the protein non-functional even if the primary structure is conserved. The second organization level is the secondary structure, which is due to the interaction between the atoms of the backbone resulting in local-folded structures.

The most common type of secondary structures are alpha-helices, beta-sheets and in addition to them there is the random coil (Which is not a true secondary structure, but is the class of conformations that indicate an absence of regular secondary structure). The third level of organization is called tertiary structure and refers to the specific shape acquired by a polypeptide chain in the 3D space. It is mainly the result of side-chain interactions of the charge-charge type, hydrophobic interactions, disulfide bonds and Van der Waals' forces[2]. And finally the quaternary structure, which is the association of several protein chains or subunits into a closely packed arrangement.

Large-scale sequencing techniques have produced gigantic and unprecedented amounts of protein data in the past period and as a direct

consequence of this massive influx, the number of protein sequences in the UniProt database[3] has surpassed 225 million as of January 2022. Only around 53 thousand of those sequences possess at least one three-dimensional structure in the Protein Data Bank (PDB)[4] Whereas the remaining majority has been left up for prediction. One of the most used approaches for protein tertiary structure prediction is comparative modeling[5], but in order to do that, a template structure with a degree of sequence similarity above a known threshold to the target protein (~>30%) is needed. When a fit template is unavailable for whichever reason, fold recognition and ab-initio methods can be used as well. Both of these approaches can benefit greatly from constraints given and laid out by the secondary structure, which gives vital importance to the secondary structure even in the determination of the organization of a higher order.

The earliest techniques developed for secondary structure prediction were established on single amino acid propensities[6] or propensities of amino acids in a window around the central residue[7]. Based on the former is the Chou-Fasman, which is about the calculation of relative frequencies of each residue to be in a given secondary structure conformation based on known protein structures. Starting from these frequencies, a set of parameters are defined and utilized to predict local secondary structure motifs of a given protein with an initial accuracy of about 50-60%. Meanwhile, the approach using the propensities of amino acids in a window around the central residue, which is known as GOR, holds an accuracy of around 60% and has been particularly popular because of its simple implementation and also because it puts the neighboring residue context in consideration. However, it was before the epoch of larger data and machine learning algorithms which were utilized and their accuracy exceeded 70%[8].

The focus of this study is to implement an approach for protein secondary structure prediction. Elaborately, a comparison was made between (GOR) and (SVM) in order to better understand and exploit the benefits of the machine learning approach. The SVM model performed slightly better on the blind test set.

2 Materials and Methods

2.1 The training set

As to begin, for the purpose of training the aspired models, a dataset which is called JPRED4[9] shall be used. Its starting set comprised 1987 representative domain sequences from each superfamily in SCOP v2.04[10]. The assumption that we will build on is that two different superfamilies should have quite different structures, so in this dataset, for each SCOP superfamily, only one representative has been extracted; this would reduce the likelihood of finding sequence similarities that are easily detectable. This dataset is also refined to be a very good quality one. Additional constraint were also used as filters were implemented in which:

i- The ones that have low resolution (i.e.: 2.5 Å) shall be getting rid of.

ii- A length limit was introduced: (i.e.:Sequence length >30 -unlikely represent protein domains- and <800 to avoid time-consuming PSIBLAST[13] runs).

iii- Missing Define Secondary Structure of Proteins (DSSP)[11] information for more than nine consecutive residues.

iv- Other filters, such as fragments removed and accession mapping inconsistencies to name a few.

In order to better grasp the diversity inside our dataset and understand its complexities, we need to perform some analyses on it.

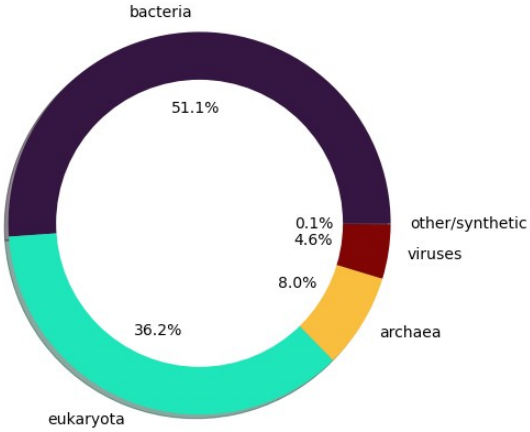

Figure 1: Pie plot showing the taxonomic composition of the dataset by super kingdom in the training set.

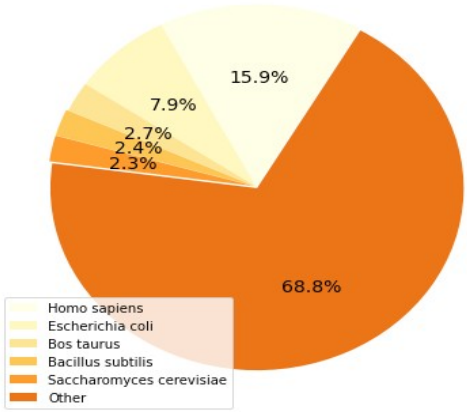

Figure 2: The figure demonstrates the largest five groups of species, and the rest are all classified under "Other".

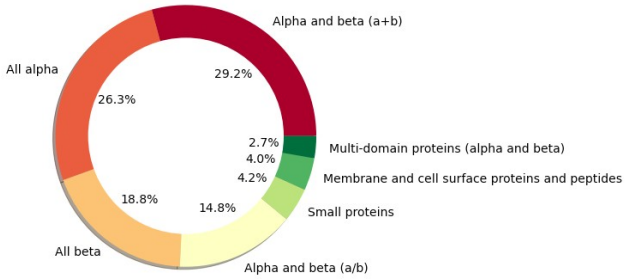

Figure 3: The composition of the dataset by SCOP classes.

As we see in figure 1, around 51% of the proteins have their origin labeled as "bacteria", 36.2% comes from Eukaryota, 8% belongs from Archea and 4.6% derives from viruses. In figure 2, in terms of species represented. Homo Sapiens come first followed by E. Coli and Bos Taurus respectively, while the remainder outside the top five are 68.8% and labeled as "Other". Looking at figure 3, shows the distribution of SCOP classes within the dataset. The three largest classes are "alpha and beta", "all alpha" and "all beta".

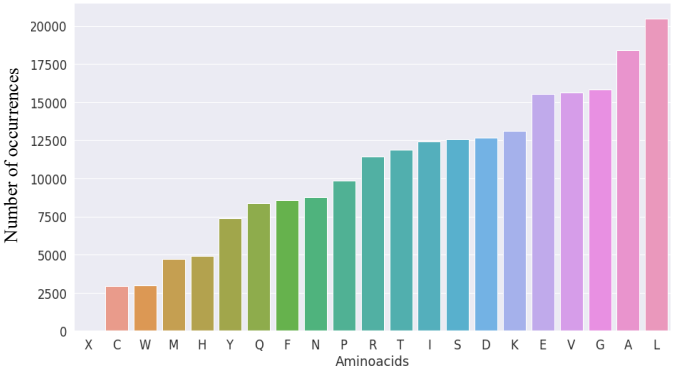

Figure 4: Bar plot that shows the amino-acidic composition of the training test dataset, obtained using the pandas package.

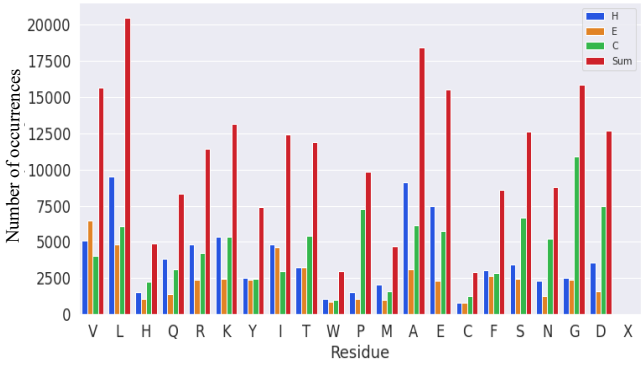

Figure 5: Bar plot that shows residues occurrences across the whole training dataset with respect to the secondary structure conformations.

2.2 Creating the blind test set

As for the blind test set, we will create one by ourselves from scratch for the purpose of this project. To thoroughly evaluate the performance of our approach, sequences were selected from PDB in September 2021 according to a specific criteria which are:

i- A deposit date which is no earlier than January 31st, 2015.

ii- X-ray diffraction as an experimental method.

iii- Data collection resolution of 2.5 Å.

iv- Only sequences ranging from 50 to 800 amino acid sequence length.

v- Inserting "Protein (only)" option in "Polymer entity Types" menu.

Using these criteria, 26839 protein IDs were obtained, and their primary sequences then were retrieved in fasta format for all their IDs' chains from PDB as well, then we remove again any retrieved fasta sequences that are not 50 to 800 amino acid residues in length and with filtering again for any potential RNAs that accompany this curent dataset. Remaining at this step at 29519 sequences.

At the next step, clustering is done to reduce internal redundancy, we clustered the sequences using MMSeqs[12] with coverage 0.5 and minimum sequence identity of 0.3 (which means if two sequences share more than 30% of sequence identity with a coverage of more than 50%, they will be grouped in the same cluster). The clustering was done via single-linkage clustering. After this, only the representative of each cluster shall be kept.

In the next step, blastp[13] was used to reduce the external redundancy, which involves removing all sequences in the blind test set that have one or more sequences match sequence in the training set. Reducing any existing similarity between the blind and training set in order for the two sets to be as independent from each other as possible. This approach was done firstly by constructing a database using the training set as an input. Next, blastp was used to search for matches between the blind set and the database using a 0.01 e-value threshold, retaining only the sequences with no matches between the two datasets. Retaining only the sequences that have higher than 30% sequence similarity. Only 309 IDs were retained from which we took 150 on random.

Further progressing, we downloaded their respective 150 PDB files and getting subsequently the 150 DSSP files from them using the "mkdssp" command (DSSP is a program used to compute secondary structure assignments from PDB files and a database storing pre-calculated secondary structure assignments). Finally, secondary structure strings and primary sequence were extracted from DSSP (only for chains selected at the step of filtering out all chains having at least one BLAST hit with SI >=30% with any sequence in JPred4 dataset).

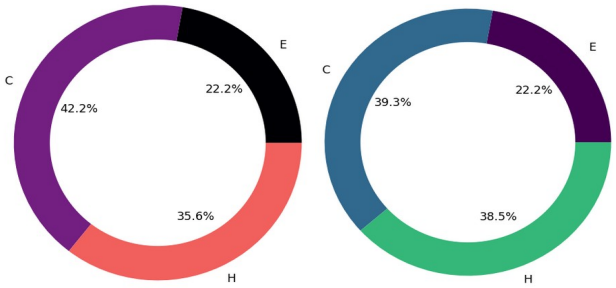

Figure 6: The distribution of secondary structure conformation in the training set (left) compared to the compositions in the blind test set (right).

2.3 Sequence profile generation

Structure is quite more conserved than sequence. Moreover, proteins with more than ~30% sequence identity over 100 aligned residues have similar structures, and we can study structurally or functionally important residues of a protein looking at conserved positions in a multiple sequence alignment (MSA). This means that a multiple sequence alignment of homologous proteins paved the way to much more information about structure than single sequences alone. explicitly, a sequence profile derived from an MSA is a compact representation of protein family that defines (for each position) which residues can be substituted by which, reflecting important constraints posed by evolution. Secondary structure prediction methods benefit greatly from this evolutionary information. While using only plain sequences is possible, complementing it with evolutionary information actually allows us to achieve much more accurate predictions at no cost virtually.

PSSM files were generated using PSIBLAST from the BLAST package for each sequence in both training and blind test sets. For convenience, the search was performed against the UniProtKB/SwissProt database, with the E-value threshold set to 0.01 and the maximum number of iterations set to 3. Sequence profiles were extracted from the PSSM files. Since the GOR uses a probabilistic framework to operate, the frequencies were normalized in the range 0-1 by simply dividing all values by 100. We also excluded the profiles that filled with all zeros and the result was 1204 profiles for the training set.

2.4 GOR method

The GOR (Garnier–Osguthorpe–Robson) technique is one of the most common secondary structure prediction methods and here it was implemented in Python language; it predicts secondary structure from an amino acid sequence using both information theory and Bayesian statistics. The evaluation of the context for each point of the sequence is the key improvement. In fact, it is based on the extension and definition of an Information Function (Equation 1) spanning windows of residues ranging from 8 residues before to 8 residues after a central location. This is made possible by the independent assumption which asserts that context residues contribute to the central residue conformation statistically and independently. As a result, there must be no association between residues found at different places in the 17-residue window.

$$I(S;R) = \log \frac{P(S \mid R)}{P(S)} \qquad (1)$$

In which:

• ***R*** refers to one of the 20 amino acids.

• ***S*** refers to one of the three secondary structure conformations.

• ***P(S)*** is the marginal probability of observing one of the secondary structure conformations.

• When a residue ***R*** is present, ***P (S|R)*** is the conditional probability of seeing a secondary structure conformation S.

If observing ***R*** and ***S*** are dependent events => ***P(R,S) != P(S)P(R)*** and:

- ***I(S;R) > 0*** ----> ***R*** is prone to be in conformation ***S***.

- ***I(S;R) < 0*** ----> ***R*** is not prone to be in conformation ***S***.

We can write (according to the notion of conditional probabilities):

$$P(S \mid R) = \frac{P(S,R)}{P(R)} \qquad (2)$$

In which:

• ***P(S, R)*** is about the joint probability of observing the conformation ***S*** and residue ***R***.

• ***P(R)***: the marginal probability of observing a residue type ***R***.

Given that the joint probability of observing the events ***S*** and ***R*** is :

$$P(S,R) = \frac{f(S,R)}{N} \qquad (3)$$

P(R) and ***P(S)*** are the likelihood of observing a residue ***R*** and the probability of observing a structure ***S*** respectively:

$$P(R) = \frac{f(R)}{N} \qquad (4)$$

$$P(S) = \frac{f(S)}{N} \qquad (5)$$

In which:

• ***f(R)*** is the total number of residues R.

• ***f(S)*** is the total number of residues observed in the conformation S.

• ***f(S, R)*** is the number of residues R observed in the conformation S.

• ***N*** is the sum of the total number of amino acids in the database.

GOR relates residues in the window to the central-residue conformation by computing an information function for each secondary structure confirmation as the sum of the single-residue functions. The prediction of the three conformations is the one with the highest value of the window-based information function Equation (1) for one of the conformations S. We can calculate the expected conformation using a 17-residue symmetric window centered at a given residue ***R*** at position ***j*** as the following:

$$S^* = \arg\max_s I\left(S; R_{-8}, .., R_8\right) \qquad (6)$$

$$S^* = \arg\max_s \sum_{k=-8}^{k=8} I\left(S;\ R_k\right) \qquad (7)$$

2.4 SVM method

SVMs are mathematical objects that can handle non-linearly separable problems by finding a hyperplane dividing two classes that meets the greatest margin requirements (two parallel hyperplanes allowing the maximum distance between the decision boundary and the nearest class points in the space).

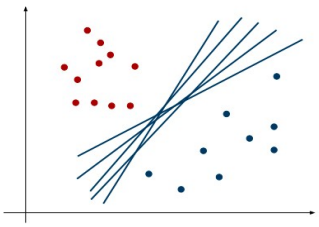

Figure 7: The process of selecting an optimal separating hyperplane[14].

A hyperparameter is defined as the set of points ***x*** that have a fixed projection on a perpendicular vector ***w***:

$$W^T x + b = 0 \qquad (8)$$

Where W is a vector that is perpendicular to the hyperparameter, and b determined on a support vector with:

$$w = \sum_{i}^{n} \alpha_i y_i X_i \qquad (9)$$

Maximizing a margin that is:

$$m = \frac{2}{\|W\|} \qquad \textbf{(10)}$$

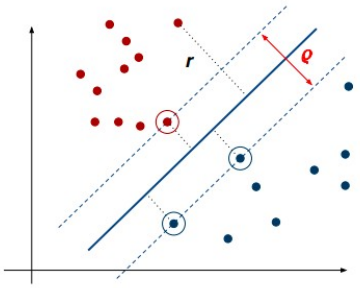

Figure 8: Optimality criterion: choosing the hyperplane which maximizes the margin, this implies that only support vectors matter, other training examples are ignorable.

Let x1, ..., xn be our data set and let yi ∈ 1, −1 be the class label of xi for y=1 and y=-1 respectively:

$$W \times X_i + b \geq +1 \qquad (11)$$

$$W \times X_i + b \leq -1 \qquad (12)$$

The following constraint-optimization problem can be used to find the decision boundary:

Minimization of:

$$\frac{1}{\|W\|^2} \qquad \textbf{(13)}$$

Subject to:

$$y_i \times \left(W^T x + b\right) \geq 0 \qquad (14)$$

for ∀***yi***

To optimize a quadratic function with linear constraints, choose the hyperplane that maximizes the margin. That is why we introduce the maximizing of the dual problem's goal function.

$$\max L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_j < x_x x_j > \qquad (15)$$

Subject to:

- $\alpha_i \geq 0$
- $\sum_{i=1}^{n} \alpha_i y_i = 0$

We can utilize a soft margin classification if the training set is not linearly separable: Slack variables ξi can be used to allow for the misclassification of difficult or noisy samples, resulting in a soft margin. Slack variables will be incorporated into the prior hard-margin formula:

$$\frac{1}{\| W \|^2} + C \sum_{i=1}^{n} \xi_i \qquad (16)$$

Subject to:

- $y_i \times \left(W^T x + b\right) \geq 1 - \xi_i$
- $\xi_i \geq 0, \forall i$

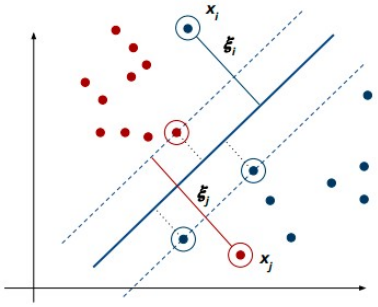

Figure 9: *In case t*he training set is not linearly separable, slack variables ξ_i can be added to allow misclassification of difficult or noisy examples. The resulting separation margin is called soft.

To minimize overfitting, the hyper-parameter C was introduced: it functions as a trade-off between the necessity of decreasing mistakes and maximizing margin. The dual problem is identical to the separable case with soft borders, but it is now subject to $0 \leq \boldsymbol{\alpha i} \leq C$. Mapping data to a higher-dimensional space is another approach to accomplish non-linearly separable classification: The concept is to utilize a feature-space modification to make a non-linearly separable issue linearly separable.

$$x \Rightarrow \varphi(x) \qquad (17)$$

We use the Kernels functions in this case, which allow us to apply such transformations to the original dataset. In some feature space, the kernel can be expressed as scalar products.

$$K(x1, x2) = \varphi(x1)\varphi(x2) \qquad (18)$$

Thus, by using the kernel, we may solve the identical issue in a different feature space without having to compute each transformation individually.

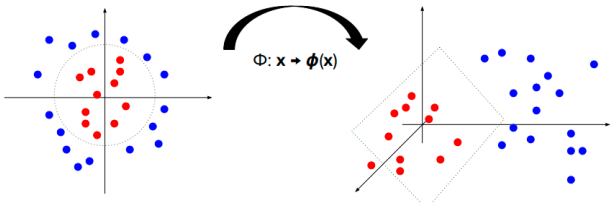

Figure 10: *The original feature space can always be mapped to some higher-dimensional feature space where the training set is separable.*

In this project, the SVM model was created using the scikit-learn module in Python[15] and it is based on numpy, scipy and matplotlib. Each dataset window was linearized into a vector with components and used as an example, along with its core residue's secondary structure. The kernel utilized was a radial basis function (RBF). The RBF kernel coefficient γ (which defines how far the influence of a single training example extends) and the penalty parameter C of the error term (which governs the trade-off between training classification accuracy and margin size) were both tuned using a grid-search approach. The best C and γ, respectively, were chosen from certain ranges.

2.5 Scoring measures of performance and evaluation

The evaluation process is a very crucial step in order to evaluate the models that were created and how they manage to predict, generalize and perform on novel data. And thus, the next step should be to test these models. In order to perform that, some scoring indices shall be considered which are used quite often in evaluation processes. In the beginning as we mentioned, the training set was the keystone to build the models in both GOR and SVM.

Hence we find a way of defining a cross validation class so that it takes into consideration a pre-defined split and this means that then you can incorporate this on a read-search object, which can be used to perform the actual read search for different values for the two hyperparameters. And these values will be tuned to see which combination of values gives the most optimal prediction in a grid search. The cross-validation is done in the following way: the 1204 full training profiles shall be separated into five sets: four sets will be used to train the model to predict the secondary structure conformation of the fifth. The cross-validation step is important to overcome the issue of overfitting.

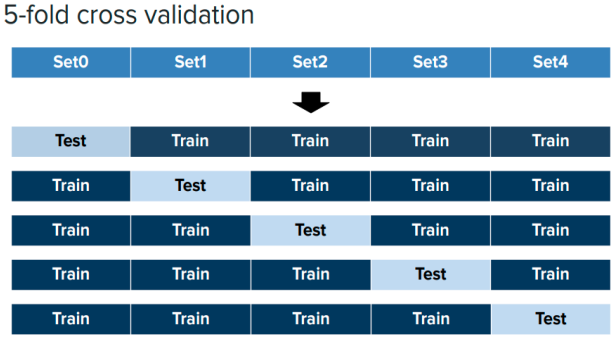

Figure 11: The cross-validation scheme implemented for both GOR and SVM.

Next, in the upcoming step after cross-validation, one needs to choose the hyperparameters that score highest in certain scoring indices and try to minimize validation set error before proceeding to test the model using the blind test set that we created from scratch. The scoring indices used in the evaluation process (for each one of the three secondary structure conformations) are:

i- Q3, which computess the overall accuracy by summing the total correct (true positive) predictions and dividing by the total. Binary matrices for each type of secondary structure can then be made from the three-class matrix to calculate the next three measures.

$$Q_3 = \frac{p_{HH} + p_{EE} + p_{CC}}{N} \qquad (19)$$

ii- Matthews' Correlation Coefficient (MCC): Which can be described as the value that indicates how strongly related the predictions to the real class.

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FP) \times (TN+FN)}} \qquad (20)$$

iii- Sensitivity: A measure of the proportion of true positives which are correct.

$$\text{Sen} = \frac{TP}{TP + FN} \qquad (21)$$

iv- Positive predictive value (PPV): A measure of the proportion of positive and negative results.

$$PPV = \frac{TP}{TP + FP} \qquad (22)$$

3 Results

The GOR method is trained/tested in the 5-fold cross-validation and mean scoring indices computed with standard errors (equation 23) and a GOR model is also trained on the complete training set and tested on the blind test set. Scoring measures on the blind set are also computed (table 2 for the cross-validation procedure, and table 3 for the blind test set).

As for SVM, as was previously mentioned, A minimal grid search procdure is run to optimize SVM parameters. This required running four independent cross-validations. The final cross-validation results for SVM are the ones achieved by the best combination of hyperparameters (C, **γ**). Scoring indices are reported as well as associated standard errors Next, using the same set of optimal hyperparameters, a new SVM model is trained on the complete training set and tested on the blind set. Scoring measures are computed as well (table 1).

$$SE = \frac{\sigma}{\sqrt{n}} \qquad (23)$$

In which:

- **σ** is standard deviation.
- n is the number of samples, in this case equals five.

SVM grid search	γ = 0.5 C=2	γ = 2 C = 2	γ = 0.5 C = 4	γ = 2 C = 4
Performance (H)	SEN= 0.700 ± 0.012 PPV= 0.845 ± 0.006 MCC= 0.660 ± 0.007	SEN= 0.128 ± 0.006 PPV= 0.877 ± 0.013 MCC= 0.254 ± 0.018	SEN= 0.688 ± 0.010 PPV= 0.838 ± 0.005 MCC= 0.646 ± 0.008	SEN= 0.129 ± 0.003 PPV= 0.870 ± 0.018 MCC= 0.252 ± 0.001
Performance (E)	SEN= 0.404 ± 0.013 PPV= 0.790 ± 0.012 MCC= 0.490 ± 0.016	SEN= 0.021 ± 0.003 PPV= 0.815 ± 0.033 MCC= 0.106 ± 0.010	SEN= 0.398 ± 0.010 PPV= 0.790 ± 0.009 MCC= 0.485 ± 0.006	SEN= 0.022 ± 0.006 PPV= 0.803 ± 0.033 MCC= 0.109 ± 0.017
Performance (C)	SEN= 0.877 ± 0.008 PPV= 0.628 ± 0.007 MCC= 0.497 ± 0.004	SEN= 0.984 ± 0.001 PPV= 0.443 ± 0.002 MCC= 0.156 ± 0.005	SEN= 0.873 ± 0.003 PPV= 0.620 ± 0.004 MCC= 0.484 ± 0.004	SEN= 0.983 ± 0.002 PPV= 0.442 ± 0.001 MCC= 0.155 ± 0.001
Q3	0.710 ± 0.002	0.468 ± 0.003	0.703 ± 0.003	0.467 ± 0.001

Table-1-: Performance of SVM: this table has been generated with the average values of the training set in cross validation.As was observed, the best prediction is the one obtained with the hyperparameters of the combination of γ0,5 and C2.

Cross-validation performance	GOR	SVM
Performance (H)	SEN= 0.803 ± 0.005 PPV= 0.634 ± 0.014 MCC= 0.526 ± 0.003	SEN= 0.700 ± 0.012 PPV= 0.845 ± 0.006 MCC= 0.660 ± 0.007
Performance (E)	SEN= 0.706 ± 0.007 PPV= 0.488 ± 0.010 MCC= 0.442 ± 0.005	SEN= 0.404 ± 0.013 PPV= 0.790 ± 0.012 MCC= 0.490 ± 0.016
Performance (C)	SEN= 0.436 ± 0.004 PPV= 0.801 ± 0.004 MCC= 0.417 ± 0.003	SEN= 0.877 ± 0.008 PPV= 0.628 ± 0.007 MCC= 0.497 ± 0.004
Q3	0.626 ± 0.001	0.710 ± 0.002

Table-2-: Demonstrates the averages performance values in cross validation and the results for the two methods are compared.

Blind set performance	GOR	SVM
Performance (H)	SEN= 0.725 PPV= 0.672 MCC= 0.496	SEN= 0.743 PPV= 0.889 MCC= 0.713
Performance (E)	SEN= 0.715 PPV= 0.475 MCC= 0.432	SEN= 0.655 PPV= 0.855 MCC= 0.690
Performance (C)	SEN= 0.468 PPV= 0.734 MCC= 0.405	SEN= 0.883 PPV= 0.682 MCC= 0.603
Q3	0.622	0.778

Table-3-: The performance of the two methods on the blind test set.

4 Conclusion and discussion

The GOR and the SVM models both showed to be relatively good predictors, achieving ~62% and ~78% of the Q3 score, respectively. SVM's improved performance, on the other hand, is closely tied to an increase in time complexity and this enhancement can also be explained by the fact that SVM condenses the information using support vectors and it avoids to consider uninformative patterns. While the GOR model's training phase took only a few seconds, the SVM model's required days. The SVM method showed a high sensitivity in predicting helices and coils conformations but the strand values were estimated quite low. MCC for was significantly higher for helices compared to the other two conformations while PPV was significantly lower for coils than the other two conformations.

In future aspects, considerable advancements might be achieved for example by incorporating the physico-chemical residue attributes as input features hence rendering the predictions more accurate with regards to giving it more biological sense. Nevertheless, provided the vast and quite successful application of machine learning approaches in this arena, it is probable that protein structure prediction will continue to exist as an active area of research and development and research in it will not reach a plateau anytime soon.

Acknowledgements

Thanks to all professors and tutors who took their time to answer questions and to my colleagues who always engaged in stimulating discussions which have often led to new perspectives, conclusions and realizations.

References

1- S. Jones, J. M. Thornton. Principles of protein-protein interactions. PNAS. (1996);93(1):13-20.

2- L. R. Engelking. Textbook of veterinary physiological chemistry. Academic Press. (2014).

3- Nucleic Acids Research. The UniProt Consortium (2017). Uniprot: the universal protein knowledgebase. (2017);45(1):158–169.

4- H. Berman, K. Henrick, H. Nakamura, J. L. Markley. The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data. (2007);35(1):301–303.

5- A. Sali, T. L. Blundell. Comparative Protein Modelling by Satisfaction of Spatial Restraints. Journal of Molecular Biology. (1993);234(3):779–815.

6- P. Y. Chou, G. D. Fasman. Conformational parameters for amino acids in helical, beta-sheet, and random coil regions calculated from proteins. (2017);13(2):211-222.

7- J. Garnier, D. Osguthorpe, B. Robson. Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. Journal of Molecular Biology. (1978);120(1):97–120.

8- B. Rost, C. Sander. Combining evolutionary information and neural networks to predict protein secondary structure. Proteins: Structure, Function, and Bioinformatics. (1994);19(1):55–72.

9- A. Drozdetskiy, C. Cole, J. Procter, G. J. Barton. JPred4: a protein secondary structure prediction server. (2015);43(1):389-394.

10- N. K. Fox, S. E. Brenner, J. Chandonia. SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. (2014);42(1):304–309.

11- W. G. Touw, C. Baakman, J.. Black, T. A. H. te Beek, E. Krieger, R. P. Joosten, G. Vriend. A series of PDB-related databanks for everyday needs. (2015);28(1):364-368.

12- M. Hauser, M. Steinegger, J. Söding. MMseqs software suite for fast and deep clustering and searching of large protein sequence sets. Bioinformatics. (2016);32(9):1323–1330.

13- S. F. Altschul, W. Gish, W. Miller, E. W. Myers, D. J. Lipman. Basic local alignment search tool. Journal of Molecular Biology. (1990);215(3):403–410.

14- S. R. Gunn. Support vector machines for classification and regression. ISIS technical report. (1998);14(1):5-16.

15- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. (2011);12(1):2825-2830.