Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: 14/12/20
Manuscript Category

OXFORD

GOR and Support Vector Machine methods: a comparative analysis in protein secondary structure prediction framework

Alessia Campo [1,*]

[1] Department of Bioinformatics, University of Bologna, Bologna, Italy

Abstract

Motivation: Protein function is strictly related to the way in which proteins acquire their native-tridimensional conformation and the determination of protein structural features became a crucial step to understand the heterogeneous nature of protein function. However, the large gap between the increasing number of known sequences and the lack of determined structures cannot be reduced by only means of costly experimental methods. To overcome this discrepancy, different computational methods have been developed and a strong focus has been placed on the secondary structure prediction framework which can be considered as an intermediate step for the 3D structure determination. In view of that we gone through an objective comparison between the GOR and SVM performances as protein secondary structures predictors analysing the different behaviour of the two methods.
Results: GOR and SVC models were built using a dataset of protein sequences of known 3D structures. Their predictor ability was evaluated and measured on an unseen protein sequences dataset.The SVC model results to be more accurate in the prediction with a three-class accuracy value of 71.2 % compared with the GOR performance that reached the 62.5 % of accuracy. Both performances were improved by the use of sequence profiles in place of single linear sequences.
Contact: alessia.campo@studio.unibo.it
Supplementary information: Supplementary materials are available at `https://github.com/AlessiaCampo/LB2_PROJECT`

1 . INTRODUCTION

The biological meaning of proteins is often associated with their ability of being involved in many cellular processes such as metabolic, structural and regulatory ones. Their ubiquity and their fascinating biological complexity has encouraged the investigation of structural and functional features from different level of protein organization.The simplest protein structural level is the primary structure, a sequence of amino acids organized in a polypeptide chain. Proteins ability to undertake different conformations and consequently different functions resides on the linear sequence information. While the prediction of native tridimensional structures from linear sequence remains quite challenging, the secondary structure prediction results in being quite successful in terms of accuracy and reliability. Protein secondary structures are determined by local folded structures that arises by the formation of hydrogen bonds between the carbonyl oxygen of one amino acid and the amino hydrogen of another.The most common secondary structural conformation are α-helices and β-strands. Random coils usually indicate the absence of regular secondary structural pattern within the protein. Since the knowledge of protein secondary structure provides useful information for the identification of protein functional domains and contributes to the improvement of fold recognition and ab initio prediction techniques, over the past decades the secondary structure prediction became one of main subject for the development of several computational methods which are intended to provide less costly and time-consuming tools to assess protein structures and functions despite the reliable but resource-intensive experimental methods such as X-ray crystallography and NMR for the prediction of the 3D structure.

Many secondary structure prediction methods were introduced between 1960s and 1970s such as the ones based on statistical information. The Chou-Fasman method (Prevelige and Fasman, 1989) is based on the computation of relative frequencies of each residue to be in a given secondary structure conformation based on known protein structures. Starting from these frequencies, a series of parameters are defined and used to predict local secondary structure motifs of a given protein sequence with an accuracy of about 50-60 %. Another notable information theory-based method is the GOR (Garnier *et al.*, 1978) method which exploits the probabilistic technique of Bayesian inference. It considers both the probability parameters as the propensities of individual amino acids to be in a given conformation and the conditional probability of a particular amino acid to form a secondary structure given that its neighbors have already formed that structure. The GOR method results to be more accurate (about 65 % of reported accuracy) than the Chou-Fasman, also because

it considers the neighbour residue context. Over time, the accuracy and the reliability of most protein structure prediction methods has grown considerably thanks to the development of more sophisticated approaches as the ones that foresees the application of machine learning methods. At the end of 1980s, multi-layered neural network (NN) and evolutionary information were introduced to this end (Bohr *et al.*, 1988). A two-layered feed-forward NN method combined with the use of multiple sequence alignments increased the prediction accuracy up to 70 % (Rost and Sander, 1994). PSIPRED (Jones, 1999) and JPRED (Drozdetskiy *et al.*, 2015) are some of the most known programs for secondary structure prediction based on NNs.The former is based on the application of two feed-forward NNs by the use of sequence profiles. JPRED makes secondary structure, coil-coiled regions and solvent accessibility predictions and it is based on the JNet algorithm (v.2.3.1). It takes in input multiple sequence alignments and consists on the application of two cascading NNs. Both PSIPRED and JPRED report an accuracy of about 81 %; also Support Vector Machine (SVM) have demonstrated to be a powerful tool for multi-class secondary structure prediction reporting a performance that is quite similar to the PSIPRED and JPRED ones (Ward *et al.*, 2003).

For the purpose of our work, we choose to implement and compare the GOR and the SVM method performances for the secondary structure prediction. We found out that the use of an SVM classifier trained and then tested on unseen data is more accurate and precise in predicting secondary structure conformations, confirming the high potentiality of machine learning approaches when applied to the protein structure prediction framework.

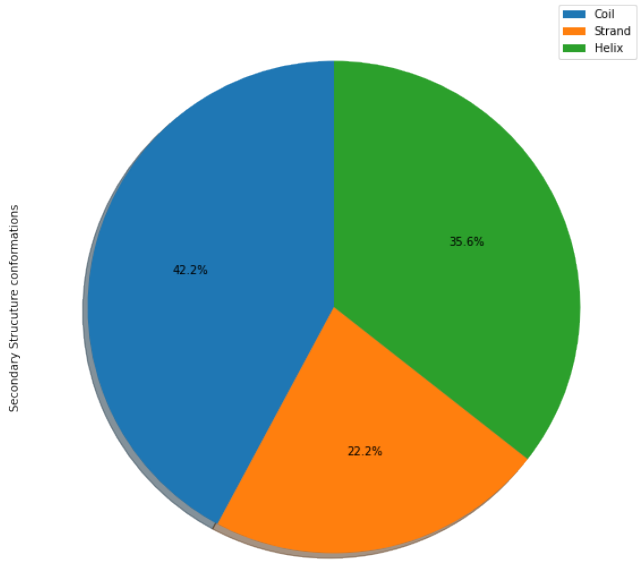

Fig. 1. JPred4 secondary structure relative abundance

2 . Materials and Methods

2.1 Training dataset description

2.1.1 Generation procedure

For the training purposes of our models, we selected the JPred4 training dataset(available at: //www.compbio.dundee.ac.uk/jpred4/index.html). It is generated from 1987 sequences, each one representative of a SCOP superfamily (v.2.04)(Andreeva *et al.*, 2014) so as to reduce sequence similarities within the whole dataset. Then the following filters were applied: (1) sequences with resolved 3D structure resolution greater than 2.5 Å, (2) sequences less than 30 residues in length were removed (reasonably choice to discard those sequences that less likely are representative of protein domains), (3) sequences longer than 800 residues were removed to make the PSIBLAST profiles generation less time-consuming, (4) sequences with missing DSSP information for more than 9 consecutive residues were also removed. Further filters were applied also checking for pairwise sequence redundancy and by removing those sequences that failed to produce PSIBLAST hits. Eventually the dataset was split into training and testing set, ending up with a training set made of 1348 sequences. The final blind test set was not used in our analysis but generated from scratch.

2.1.2 Statistical analysis

In order to better demonstrate the reliability and the adequacy of our training dataset we performed some statistical analysis that highlight different crucial aspects of the protein selection. We firstly focused on the secondary structure conformations distribution across the JPred4 dataset (**Fig. 1**). We assessed the comparative residue composition of the whole dataset and of the fraction of helices, strands and coils residues (**Fig. 2**)

From Figure 1 it is possible to see how the dataset is quite balanced as concerns the secondary structures occurrences across the whole dataset with a bias on helices respect to strands. Figure 2 shows the relative abundance of each residues in the overall dataset and within a specific secondary structure conformation. The plot highlights the single residue

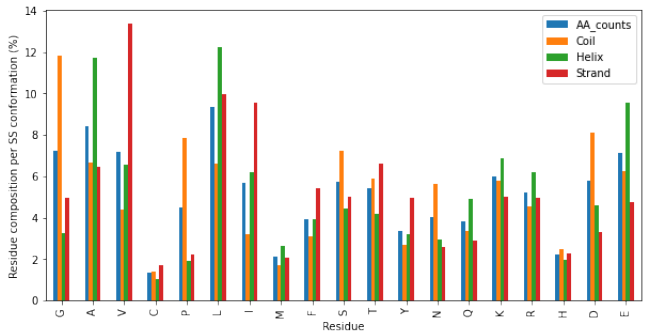

Fig. 2. Residue composition in the dataset and in relation to each secondary structure conformation

propensity to be in a given conformation: Alanine (A) and Leucine (L) have high helix-forming propensities, whereas those residues that tends to disrupt the helices for their rigidity as Proline (P) or for their high flexibility as Glycine (G) are less frequent in that conformation. Valine and Isolecuine are most common in strands as the other aromatic amino residues (W, T and F). It is also possible to compare these results with the one reported by the Chou-Fasman propensity scale on ProtScale (Gasteiger *et al.*, 2005).

Detailed statistics analysis led us to report the taxonomic classification at the level of Superkingdom (**Fig.3**). Comparing the proportions shown in Figure 3 with the ones reported in the statistics section of the Uniprot Database (UniProt, 2019) (release 2020_10), it is possible to highlights the fact that our dataset properly represents the real protein space. Figure 4 shows the distribution of secondary structure classes within the JPred4 dataset according to the SCOP classification. It shows how the most abundant one is the mixed *alpha+beta* class. This proportion allows us to realize how the dataset covers a wide spectrum of different fold conformations.

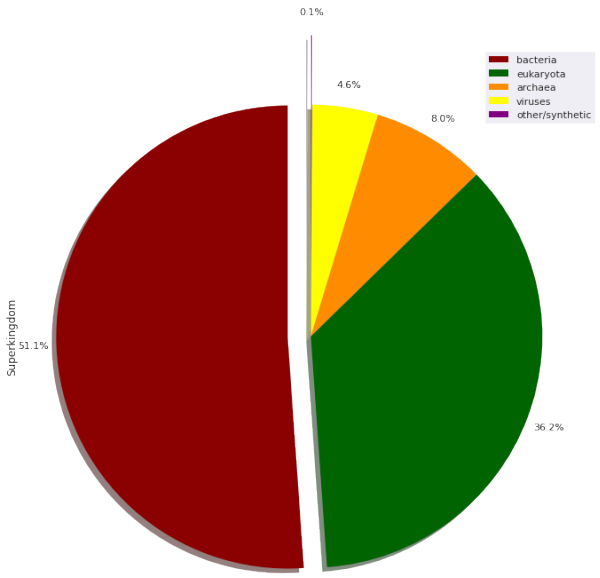

Fig. 3. JPred4 Superkigdom distribution

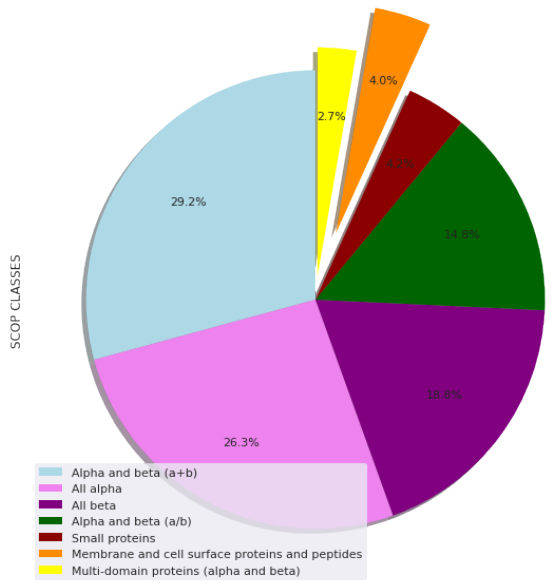

Fig. 4. JPred4 secondary structure conformation distribution according to SCOP classification

2.2 Blind test set generation

2.2.1 Generation procedure

To estimate the performances of the methods for secondary structure prediction we choose to generate a blind test set that ensure the never-seen-before condition during the evaluation step. Indeed its independence from the training set allows to avoid biases in the prediction procedure. To produce a reliable and affordable test set we extracted proteins from Protein Data Bank (Berman *et al.*, 2000) (release 2020) applying the following search criteria and also relying on the training set main features to be consistent:

- Depositing after Jan, 2015 (after the JPred4 deposition in 2014)
- High-quality experimental parameters, mainly searching for X-ray crystals with resolution less than 2.5 Å
- Chain length between 50 and 300 residues
- Exception of DNA/RNA polymer entities
- Exception of engineered mutations

We retrieved 21737 structures that have met the applied criteria. After having extracted all the corresponding `fasta` files, we applied additional filter to remove any sequence with chain length less than 50 and greater than 300 residues, also removing sequences with more than two ambiguous character (X). We ended up with 48240 `fasta` sequences. A crucial step was to reduce the internal and external redundancy that results in the presence of sequences with high similarity respectively within the test set and with respect to the JPred4 training set. To reduce the internal redundancy we choose to perform a clustering procedure by means of MMseqs2 software (Many-against-Many-searching, v. 12.113e3) (Steinegger and Söding, 2017) which implements a cascaded clustering algorithm running on multiple cores. The algorithm compares all the sequences in the sequence database with each other using the `mmseqs` search and then it filters the alignments following a set of user-specified parameters: we set up a sequence identity of 30 % and an alignment coverage of 50 %. At the end the program produces clusters grouping together sequences with sequence identity higher than 30 % and coverage greater of 50 %. Therefore we kept all the representative sequences of each resulting cluster ending up with 4167 sequences. The reduction of the external redundancy ensures a stronger independence between training and test set. Thus, we produced a database with the kept set of sequences using the `makeblastdb` option from BLAST+ program (Basic Local Alignment Search Tool) (Altschul *et al.*, 1990). Using BLASTP, we aligned the training set sequences against the newly-created database and we filtered out all the sequences that produced less significant results using an e-value threshold of 0.01 and that were found to have a sequence identity greater than 30 % with any sequence of the JPred4 dataset. Ending up with 3643 proteins, we randomly extracted 150 sequences to set-up the final blind test set. In order to extract the corresponding secondary structure assignments of all the 150 selected proteins, we processed the corresponding PDB files running the DSSP (Define Secondary Structure of Proteins) program (Kabsch and Sander, 1983) that computes the most likely secondary structure assignment given protein atomic-resolution coordinates and identifies the presence of H-bonds by the calculation of the electrostatic energy between C, O, N and H atoms. Based on this, helices and strands are assigned according to the H-bond pattern identified by the algorithm. Since the DSSP output returns eight different secondary structure assignments we processed the files in order to obtain a mapping of the secondary structure conformations to three main classes: H for helices, C for coils and E for strands. Eight-conformation assignment:

- alpha-helices → H
- residues in isolated beta-bridge → B
- extended strands within beta-ladder → E
- 3/10-helices → G
- pigreco-helices → I
- hydrogen bonded turn → T
- bend(high curvature) → S
- no assignment → ""

Mapping results:

- H,G,I → H
- B,E → E
- T,S,"" → C

As result of the mapping procedure, we produced a secondary structure string for each protein of the blind test set.

2.2.2 Statistical analysis

We performed some general statistic analysis on the blind test set, focusing on the residues abundance both across the test set and in relation to the secondary structure conformation (**Fig.5**). Also the secondary structure conformations distribution are reported (**Fig.6**).

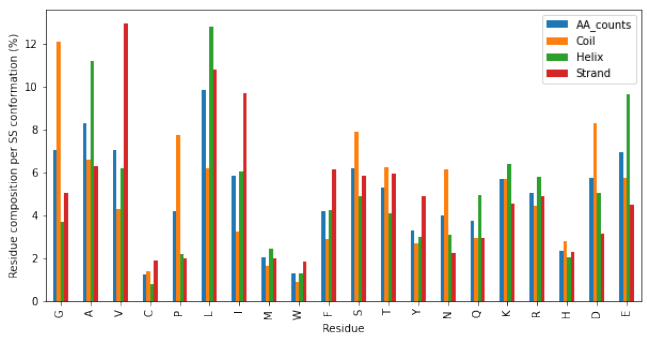

Fig. 5. Residues relative frequencies across the whole data set and with respect to the secondary structure conformations. The order of the residue across the x-axis follows the chemical-physical properties of the aminoacids

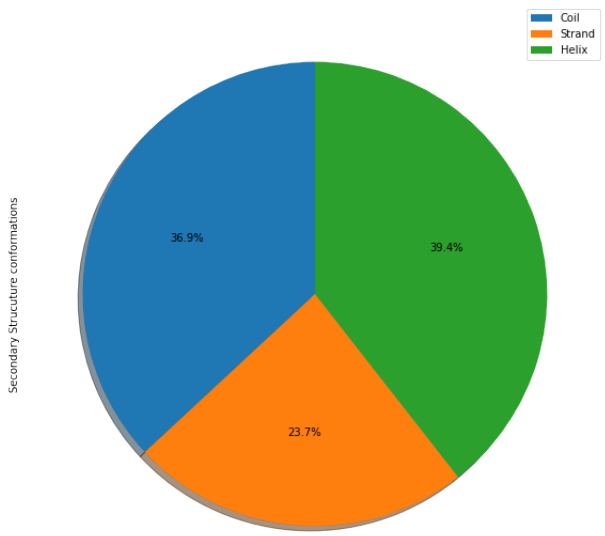

Fig. 6. Secondary structures distribution on the blindset

Figure 5 shows that the most abundant residue in the dataset are Leucine (L) and Alanine(A). Once again their occurrence in helices conformation is quite remarkable as for Valine (V) in strands. On the other hand Tryptophan (W), Hystidine(H), Metionine (M) and Cysteine (C) are poorly distributed in the overall set. Comparing these data with the ones obtained for the training set (**Fig.2**) it is possible to observe an overall coherence and likeness between the two dataset residue abundance and distribution. Even the secondary structure distribution seems to be quite similar to the one observed in the training set.

2.3 Sequence profile generation

As seen in previous studies, when using multiple sequence alignments (MSA) in place of single sequences it is possible to catch evolutionary information that improve the overall prediction performance. Indeed, scanning MSA aligned positions it is possible to detect highly conserved regions that are essential to recognize the importance of certain residues in determining structural and functional features of single proteins and of the protein family they belong. MSA are then helpful to identify a certain level of sequence variability among distantly related proteins that share similar structures and functions but that present low sequence identities.The rationale behind that relies on the observation that protein structures diverge more slowly in the evolution time than sequences. All these information can be captured by the generation of sequence profiles, matrices that list the frequencies of each residue in each position of the original MSA between a target sequence and other similar ones. For the above reasons, we generated sequence profiles for each sequence of the training and blind test set by the use of PSI-BLAST program (Position-Specific Iterative Basic Local Alignment Search Tool) (Altschul *et al.*, 1997). In the first iteration, given a query sequence, it searches against a target sequence database for regions of the most significant local alignment by the use of a substitution matrix. Then, it generates a multiple sequence alignment of the highest scoring pairs above a certain e-value threshold and compute a Position-Specific-Substituion Matrix (PSSM) that captures the conservation pattern in the alignment. The PSSM is used as scoring matrix in the further searches and updated at each iteration. After a user-defined number of iterations the program returns the PSSM and the sequence profile derived from the final MSA. For the purpose of our analysis, we generated a protein sequence database from UniProtKB/SwissProt using the option `makebalstdb`. Then we run PSIBLAST using as query the `fasta` sequences of training and blind test set. We set an e-value threshold equal to 0.01, a maximum number of iteration equal to 3 and a number of alignments equals to 1000. The output files have been filtered to remove those files lacking of PSI-BLAST hits or having empty profiles meaning that no significant alignments were found during the search. The final training set contain 1204 sequences, while the blind test set 126.

2.4 GOR method

2.4.1 Theoretical overview

As mentioned before, the Garnier-Osguthorpe-Robson (GOR) method is an information theory-based method that combined with the Bayesian statistical inference technique, takes into account not only the single residue propensity in a specific position to form a given secondary structure conformation but also the influence of neighbouring residues respect to that residue position. The original implementation has been refined and improved over the years. The most crucial change was the inclusion of evolutionary information through the use of MSA and sequence profiles (Sen *et al.*, 2005) that led to an increment of the information content and to an improvement in the prediction of secondary structure conformations. The basic formulation considers the single-residue context and derives the information function to predict the conformation of a given residue as shown in the equation (1).

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

Using the probability chain rule Equation (1) is modified as follows:

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

where $P(R,S)$ is the joint probability of observing the residue R in the conformation S; $P(R)$ and $P(S)$ are the marginal probabilities

of observing respectively the residue R and the conformation S. This information is estimated from training data and then applied on unseen protein sequences to perform prediction: the highest value computed from equation (2) for one of the three conformation is assigned to a given residues and represents the propensity of that residues R to be in that conformation S. The equations (1) and (2) can be extended to a local sequence context by considering a certain number of consecutive residues, also known as residue windows. This type of formulation allows to consider not only the single residue context, but also the neighbouring residues with respect to a central residue in the window. The formulation becomes window-based as follows:

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

where d is the position index inside the residue window. Based on previous studies about information content at increasing separation d should assume values in the range of -8 (N-terminal) and $+8$ (C-terminal). From this assumption and by also considering the central residue position (R_0) the total window size should be equal to 17. Since the computation of all the possible configuration is very expensive (20^w where w is the number of residues to be determined) and a very large database is required to estimate reliable distributions, a statistical independence assumption can be introduced: it is possible to assume that there is no correlation between residues occurring at different positions of the window. The mathematical formulation of the corresponding information function is defined as follows:

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

where $(S; R_k)$ is the single-residue based information function shown in Equation (2). The final value of the window-based information function is the sum of all the single-residue information functions inside the window. The prediction is always based on the assignment of the conformation that has the highest value:

$$S* = argmax_S I(S; R_{-d}, ..., R_d) \tag{5}$$

As stated previously, one of the major improvements of the GOR method was the use of sequence profiles in place of single sequences as input. In this case the mathematical formulation is:

$$I(S; PW) = \sum_{k=-d}^{d} \sum_{R} PW_{k,R} \times I(S; R_k) \tag{6}$$

where PW defines the residue frequency within the profile window.

2.4.2 Implementation

We followed a general workflow using sequence profiles as input by firstly implementing a python script for the training procedure: it takes in input the profile files and the dssp files containing the secondary structure strings for each sequence in the training set; for each input sequence it scans the sequence profile by generating the so called sliding windows and uses the frequencies stored in the sequence profiles of the central residue to update the initialized matrices H, E, C accordingly to the secondary structure assignment and R (based on the overall residue frequency). At the end the program returns the matrices resulting from the completed iteration over the whole training set. The four matrices stacked into one are then used as input model along with the profiles of sequences to be predicted for for a python program that computes the information function according to the Equation (6) and returns the predicted secondary structure sequences following the logic shown in Equation (5). All details about the implementations are in the supplementary materials.

2.5 Support Vector Machine method

2.5.1 Theoretical overview

Support Vector Machine (SVM) are supervised learning models used for classification tasks. Given a set of training examples and a set of two or more different classes to which each example belongs, the SVM training algorithm produces a model that is the representation of the examples as points in the space so that they are separated according to the classes to which they belong. The prediction goals of SVM consists on mapping into the same space new unseen examples that are predicted to belong to a class based on which side they fall. Formally, given a set of training elements in a d-dimensional space $x_i \in R^d$ that belong to a class $y_i \in -1, +1$ SVM builds one or more separating hyperplanes mathematically defined by the linear equation $w^T x + b = 0$ where $\vec{w}$ is the perpendicular vector to the hyperplane and b is the intercept. For each point separated by the hyperplane we define:

$$y_i(w^T x_i + b) \geq \frac{\rho}{2} \tag{7}$$

where ρ is the margin (ie the distance between the closest points belonging to different classes) that has to be maximized. Since there could be many possible linear separators, the optimality criterion is to choose the hyperplane that maximizes the margin by considering only the support vectors (critical point that lies on the margin) for which the above inequality becomes an equality: $y_i(w^T x_i + b) = \frac{\rho}{2}$. Rescaling w and b by $\frac{\rho}{2}$ we obtain the distance between a support vector x_s and the hyperplane as shown in the Equation (8)

$$r = \frac{y_i(w^T x_s + b)}{\|w\|} \tag{8}$$

$$\rho = \frac{2}{\|w\|} \tag{9}$$

Now we can formulate the quadratic optimization problem as to minimize the function $\frac{1}{2}\|w\|^2$ subject to the linear constraints $y_i(w^T x_i + b) \geq 1 (\forall i)$ that determine the correct separation of the two classes. The solution for the optimization problem involves the construction of a dual problem by the use of a Lagrange Multiplier α_i associated to each constraints $y_i(w^T x_i + b) \geq 1$ and it is described as follow:

$$maximize \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \tag{10}$$

subject to $\alpha_i \geq 0 \, \forall i$
$\sum_{i=1}^{n} \alpha_i y_i = 0$

The solution for w and b is:

$$w = \sum_{s} \alpha_i y_i x_s \tag{11}$$

$$b = y_k - \sum_{s} \alpha_s y_s \langle x_s, x_k \rangle \tag{12}$$

In the solution the non-zero α_i indicate that the corresponding x_i is a support vector. The classification function can be written as:

$$f(x) = \sum_{s} \alpha_s y_s \langle x_s, x \rangle + b \tag{13}$$

The formulation described above is suitable only for linearly separable problem. When we face with non linear separable problems , it is possible to use the soft margin classification that foresees the introduction of the slack variable ξ_i for every data point x_i so that if x_i is on the wrong side the slack variable indicates the distance between the data point and its class

margin, otherwise it is zero. For this reason ξ_i allows to model potential errors of difficult or noisy examples. The quadratic optimization function can be modified as follows:

$$minimize\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i with \xi_i \geq 0 \forall i \tag{14}$$

where C is an hyper-parameter that can be set to control the overfitting. For large values of C the margin will be smaller and the error rate in the training classification will be low, conversely a small value of C leads to a larger margin separating hyperplane and allows to a most likely misclassification of the training points. The solution of the soft margin classification problem is likewise defined by the dual problem in which we slightly modify the upper-bound of alpha as $0 \leq \alpha_i \leq C$.Also the solutions of the dual problem are quite similar except for $b = y_k(1 - \xi_i - \sum_s \alpha_s y_s \langle x_s, x_k \rangle$

However in many cases the dataset used in the analysis can be hard to be solved even with the use of soft margins. A solution to this problem is to map the original feature space into a higher-dimensional space as to make the dataset separable. As seen in the above equations for hard and soft margins the computation of the decision boundary relies on the inner product between the two vectors, so when mapping the data into a higher dimensional space, we can rely only on the inner product in that higher dimensional space. After the transformation ϕ the inner product will be:

$$K(x_i, x_j) = \langle \phi(x_i), \phi x_j \rangle \tag{15}$$

The equation above explain what a kernel function is: a function that is equivalent to an inner product in some feature space. Applying the kernel function is possible to map the data in a higher-dimensional feature space with no need to compute each transformation explicitly. Many kernel functions can be used for this purpose. The Radial Basis Function (RBF) kernel is one of the most used and is presented in the following formulation:

$$K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2) \tag{16}$$

where γ is an hyperparameter (more explanation in the next section).

2.5.2 Implementation

The purpose of our analyses was to apply an SVM method for secondary structure prediction. We used multi-class Support Vector Classification (SVC) available in the Python-based scikit-learn library (Pedregosa *et al.*, 2011). The SVC constructor can be used to define the SVC model by specifying the γ and the C parameters. The γ parameter defines how far the influence of a single training example goes. Small values of gamma means a wide-spread influence of a point x_i with a class y_i on deciding the class of another point x_j even if the distance between them is very large, vice versa if gamma is big. Technically speaking, large gamma leads to high bias and low variance models, and vice-versa. By the implementation of a python script we scanned all the sequence profiles of both the training and blind test set and we transformed them into input vectors. In particular, we built a matrix X by scanning each profile and by creating the 17-residue sliding windows. Each window is flattened in a single feature vector 340 elements long and it corresponds to a row of the matrix X. We built the vector y containing the actual class of the central residue of each window. The classes were converted from H, E, C to 1, 2 and 3. The same was done for the testing set. Also the classification report was used to visualize the performance scores of the SVC method implementation.

2.6 Evaluation procedure

2.6.1 Motivation

The main aim of our analysis is to compare the performances of the aforementioned methods applied to the secondary structure prediction task that consists on the labelling of each residue with one of the three possible secondary structure conformations. Therefore we can define our problem as a multi-class classification problem. In this context, when evaluating the models performances ++++we must consider that the GOR method foresees the computation of the information function for all the 3 conformations and assigns the most probable to each residue, while the SVC uses the one-vs-rest approach training three different binary classifiers (H vs non-H; C vs-non-C; E vs non-E). Based on the discrimination function of the three classifier it assigns the predicted class to each residue. We performed the evaluation of both methods on a testing set assuming the one-vs-rest approach, since both works by taking into account all the possible conformation and comparing the information and discrimination function to choose the best assignment++++.

2.6.2 Evaluation metrics

The scoring indexes used for the assessment of the performances were computed starting from the settlement of a multi-class confusion matrix derived from the three binary matrices (**Table 1** and **Table 2**)

		Predicted H	Predicted n-H
Actual	H	c_h	u_h
	n-H	o_h	n_h

Table 1. Here it is reported an example of binary confusion matrix for the first class H.

		H	E	C
Actual	H	p_{HH}	p_{HE}	p_{HC}
	E	p_{EH}	p_{EE}	p_{EC}
	C	p_{CH}	p_{CE}	p_{EE}

Table 2. The three-class confusion matrix

The three class accuracy in computed from the matrix by summing the number of true positives (ie correctly predicted residue) of each class and dividing them by the total number of residues.

$$Q_3 = \frac{P_{HH} + P_{CC} + P_{EE}}{N} \tag{17}$$

From the binary matrices of each class, Sensitivity, Positive Predictive Value (PPV) and Matthews correlation coefficient (MCC) are computed as shown respectively in the equations (18)(19)(20)

$$Sen_S = \frac{c_S}{c_S + u_S} \tag{18}$$

$$PPV_S = \frac{c_S}{c_S + o_S} \tag{19}$$

$$MCC_S = \frac{c_S \times n_S - o_S \times u_S}{\sqrt{(c_S + o_S) \times (c_S + u_S) \times (n_S + o_S) \times (n_S + u_S)}} \tag{20}$$

where S is one of the three classes. Sensitivity (or recall) is the proportion of true positive correctly identified; PPV (or precision) is the fraction of true positive among all the positive predictions; MCC a quality measure

of a binary classification. It takes into account true and false positive and negatives and for this reason hardly leads to biased results even in the case of class imbalance.

2.6.3 Cross-validation

Before having proceeded with the final evaluation of the GOR and SVC models on the blind test set, we performed a 5-fold cross-validation to verify the consistency of the perfomances and assess the accuracy values that tends to be unsteady respect to the datasets used. We split the original training dataset into 5 subsets and we performed a training/testing procedure leaving one subset as testing set in turn. Once the entire 5-fold cross-validation was completed, we computed the average, the standard deviation and the standard error for each scoring indexes over the five folds. For the SVC model evaluation the 5-fold cross-validation consisted in performing also a grid search of the γ and C hyper-parameters: four combinations of values were tested on each complete cross-validation and the combination that reached the best performance scores was selected for the final evaluation.

3 Results

The evaluation of the performances of the GOR and SVM methods both on cross-validation procedure and final evaluation on the blind test set are here reported. The 5-fold-cross-validation for both methods was performed using the assignment reported on the dssp files as actual classes and all the scoring indexes above described were computed. The minimal grid search performed for the optimization of the SVC hyperparameters was done on the combination of four values: $C = 2.0, 4.0$ and $\gamma = 0.5, 2.0$. Each combination was tested on a complete 5-fold cross-validation. Comparing the scoring indexes resulting from the grid search we found out that the combination $C = 2.0$ and γ=0.5 outperforms the others. We report here the scores obtained from the minimal grid search (**Table 3**).

	$MCC \pm SE$	$PPV \pm SE$	$SEN \pm SE$	Q3
C=2.0 γ=0.5	0.544 ± 0.054	0.753 ± 0.066	0.657 ± 0.137	0.706 ± 0.002
C=2.0 γ=2.0	0.172 ± 0.042	0.713 ± 0.136	0.377 ± 0.305	0.467 ± 0.003
C=4.0 γ=2	0.181 ± 0.048	0.682 ± 0.123	0.384 ± 0.299	0.473 ± 0.003
C=4.0 γ=0.5	0.52 ± 0.040	0.747 ± 0.065	0.653 ± 0.135	0.701 ± 0.002

Table 3. For each combination of hyperparameters, the table reports the overall average scoring indexes

As shown in the table above, the best combination of hyper parameters is the first one ($C = 2.0$,$\gamma = 0.5$), with the highest average MCC value equals to 0.544.The MCC value indicates that the model performs quite well on the prediction of the classes.
The scoring indexes for the evaluation of the GOR model have been computed in a similar manner (detailed information in the supplementary material, Figure 5) and have proved a strong consistency of the performance.

The final evaluation of the two methods consisted in training once more the GOR and SVC models on the entire JPred4 dataset and testing it on the blind test set. We report the results(**Table 4**), emphasizing the comparison between the two performances:

		GOR	**SVC**		**GOR** $MEAN \pm SE$	**SVC** $MEAN \pm SE$
MCC	H	0.454	0.647		0.421 ± 0.016	0.561 ± 0.045
	E	0.406	0.545	$\longrightarrow$		
	C	0.405	0.492			
PPV	H	0.598	0.860		0.632 ± 0.030	0.749 ± 0.080
	E	0.605	0.796	$\longrightarrow$		
	C	0.693	0.593			
SEN	H	0.835	0.692		0.596 ± 0.120	0.688 ± 0.098
	E	0.447	0.485	$\longrightarrow$		
	C	0.507	0.857			
Q3		0.625	0.712			

Table 4. All the scoring indexes of the final evaluation are listed above. In the first part of the table are reported the single values for each scoring indexes of a given class. The second part of the table reports the average values and the standard errors.

4 Discussion

In this work we addressed the problem of secondary structure prediction by comparing two different methods. Starting from the statistical analysis of our dataset, we have demonstrated an overall coherence of our data with respect to the real protein space. The objective comparison of the performances of the GOR and SVM methods allows us to better understand the improvement made on this field. From the results shown in section 3 it is possible to observe how the SVM approach outperforms the GOR method: MCC values for all the classes tends to be higher, in particular for helices and strands (**Table 4**). The SVC model tends to be more precise in the prediction of helices and strands accordingly to the PPV scores. Also, the sensitivity scores decrease except for coils meaning that the SVC classifier is more "permissive" in predicting that class.

5 Conclusion

Overall the SVC predictor performs better than GOR with an increment of the accuracy (Q3) up to 71,2 %. This enhancement can be explained by the fact that SVM condenses the information using support vectors and it avoids to consider uninformative patterns. Also the statistical independence assumption of the GOR method can be a limitation when considering the contribution of the sequence context to the central residue conformation. Major improvements can be done for example by processing the physico-chemical residue characteristics as input features thus making the predictions more accurate in terms of biological meaning. However,given the extensive and successful application of machine learning approaches in this field, it is likely that protein structure prediction will continue to be an active area of research and development.

References

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**(3), 403–410.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, **25**(17), 3389–3402.

Andreeva, A., Howorth, D., Chothia, C., Kulesha, E., and Murzin, A. G. (2014). Scop2 prototype: a new approach to protein structure mining. *Nucleic acids research*, **42**(D1), D310–D314.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The protein data bank. *Nucleic acids research*, **28**(1), 235–242.

Bohr, H., Bohr, J., Brunak, S., Cotterill, R. M., Lautrup, B., Nørskov, L., Olsen, O. H., and Petersen, S. B. (1988). Protein secondary structure and homology by neural networks the α-helices in rhodopsin. *FEBS letters*, **241**(1-2), 223–228.

Drozdetskiy, A., Cole, C., Procter, J., and Barton, G. J. (2015). Jpred4: a protein secondary structure prediction server. *Nucleic acids research*, **43**(W1), W389–W394.

Garnier, J., Osguthorpe, D. J., and Robson, B. (1978). Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of molecular biology*, **120**(1), 97–120.

Gasteiger, E., Hoogland, C., Gattiker, A., Wilkins, M. R., Appel, R. D., Bairoch, A., *et al.* (2005). Protein identification and analysis tools on the expasy server. In *The proteomics protocols handbook*, pages 571–607. Springer.

Jones, D. T. (1999). Protein secondary structure prediction based on position-specific scoring matrices. *Journal of molecular biology*, **292**(2), 195–202.

Kabsch, W. and Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, **22**(12), 2577–2637.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830.

Prevelige, P. and Fasman, G. D. (1989). Chou-fasman prediction of the secondary structure of proteins. In *Prediction of protein structure and the principles of protein conformation*, pages 391–416. Springer.

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

Sen, T. Z., Jernigan, R. L., Garnier, J., and Kloczkowski, A. (2005). Gor v server for protein secondary structure prediction. *Bioinformatics*, **21**(11), 2787–2788.

Steinegger, M. and Söding, J. (2017). Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, **35**(11), 1026–1028.

UniProt, C. (2019). Uniprot: a worldwide hub of protein knowledge. nucleic 944 acids res 47. *D506-D515*, **945**.

Ward, J. J., McGuffin, L. J., Buxton, B. F., and Jones, D. T. (2003). Secondary structure prediction with support vector machines. *Bioinformatics*, **19**(13), 1650–1655.

6 Supplementary

METRICS	**SS**	$MEAN \pm SE$ **PER SS**	**OVERALL** $MEAN \pm SE$
SEN	H	0.872 ± 0.002	0.611 ± 0.131
	E	0.449 ± 0.004	
	C	0.514 ± 0.002	
PPV	H	0.571 ± 0.006	0.636 ± 0.063
	E	0.576 ± 0.100	
	C	0.763 ± 0.001	
MCC	H	0.492 ± 0.001	0.438 ± 0.029
	E	0.390 ± 0.004	
	C	0.434 ± 0.001	
Q3		0.627 ± 0.002	

Table 5. Here are shown all the scoring indexes values computed from the GOR 5-fold cross-validation. In the first part the average values of all the metrics relative to each class are reported. In the second part the same scores are computed in the overall