Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Predicting secondary structure of proteins: a comparison between GOR method and Support Vector Machines

Matteo Bolner

Abstract

Motivation: The determination of protein structure is one of the most important open problems of bioinformatics and a key passage for functional annotation. As the number of protein sequences available steadily increases, the number of structures lags behind, since as of today there are still no methods for the complete and accurate prediction of the tridimensional structure of a protein starting from its primary sequence of residues. The prediction of the secondary structure could help bridge this gap; in this project, two methods for secondary structure prediction, GOR and SVM, are implemented and tested on a set of protein sequences with known structure, and compared in performance. The characteristics and potential of such methods is discussed in depth.
Results: The methods were succesfully implemented, and two models were produced. The models are able to predict the secondary structure of a protein chain with only the primary sequence as input, with a three-class accuracy of 62% (GOR) and 70% (SVM). While the accuracy of the SVM method is clearly superior, the overall difference in performance is more nuanced.
Availability: `https://github.com/matteobolner/lb2_project`
Contact: matteo.bolner@studio.unibo.it
Supplementary information: Supplementary data are available at `https://github.com/matteobolner/lb2_project`

1 Introduction

The determination of protein function is a key passage in the understanding of biological systems, and is necessary for medical research, drug design and for the improvement of all biotechnological applications of proteins.
In order to determine the function of a protein it is necessary to know its three-dimensional structure: nowadays, the most reliable way to obtain it is experimentally, usually by determining its electron density through X-ray cristallography. However, experimental methods are expensive and time consuming, which is why on the Protein Data Bank (http://www.rcsb.org [1]) there are only circa 147000 deposited protein structures, compared to the circa 561000 manually annotated and reviewed protein sequences on UniProtKB/Swiss-Prot, not to mention the 179.000.000 automatically annotated and unreviewed sequences on UniProtKB/TrEMBL (https://www.uniprot.org [2]).
One of the core open problems of bioinformatics is the development of methods for the prediction of a protein's three-dimensional structure starting from its primary structure, the linear sequence of aminoacids. While the primary structure of a protein dictates its three-dimensional structure, when taken by itself as a string of characters it is not very informative: as of now, it is not possible to use it to accurately predict the tertiary structure. The secondary structure is much more informative: the folding of local segments of the protein in α-helices, β-sheets or coils can provide critical information for the identification of functional protein domains, in addition to constraints necessary for tertiary structure prediction techniques such as homology modelling, fold-recognition techniques, ab-initio and others.
Methods for the prediction of the secondary structure have been under development since the 1960's : the so-called "first generation" methods implement simple stereochemical principles (Lim, 1974 [3]) or statistics (Chou,Fasman 1974 [4]) to try to correlate the content of the single residues with the corresponding secondary structure; for example, the Chou-Fasman method is based on the relative frequency of each amino acid in each secondary structure conformation derived from a set of solved protein structures. The parameters derived from these frequencies are used to predict local secondary structure motifs from aminoacidic sequences. The accuracy of these methods (around 50-60%) has been improved upon by "second generation" methods, which consider not only the single residue propensity but also the adjacent residues: for example, the GOR method (Garnier $et\ al.$, 1978 [5]) uses a sliding window of residues and implements information theory concepts and bayesian statistics. However, the accuracy of second generation methods is stalled at slightly more than 60%. The breakthrough came with the third generation of techniques, characterized by the usage of evolutionary information derived from the alignment of multiple homologous sequences: this way,

the conservation of structure with variation in sequence is accounted for, and additional information may be obtained from the observed patterns in sequence variability and the location of insertions and deletions. In particular, the most conserved regions in a multiple sequence alignment are usually assumed to be either functionally relevant or buried in the hydrophobic protein core, while the most variable residues are usually part of the protein surface; starting from these observations, Benner and Gerloff (Benner, Genioff 1991 [6]) demonstrated that the degree of solvent accessibility of a residue can be predicted fairly easily; this can be helpful in the prediction of secondary structure by comparing patterns of solvent accessibility. Combined with the increasingly large databases of protein structures and the advanced algorithms implemented, third generation techniques can reach more than 70% of accuracy. PSIPRED (Jones, 1999 [7]), one of the most popular predictors available, implements two feed-forward neural networks that process sequence profiles deriving from PSI-BLAST output; its multi-class accuracy can reach more than 80%. By introducing structurally similar sequences already deposited on the PDB in the prediction, the SSpro and ACCpro predictors (Magnan *et al*, 2014 [8]) were able to obtain more than 90% of accuracy in both secondary structure and solvent accessibility prediction. Jpred4 (Drozdetskiy *et al*, 2015 [9] is another well-known predictor, based on the JNet algorithm (Cuff *et al*, 2000 [10]) which implements an ensemble of neural networks and uses multiple sequence alignments, hidden Markov model profiles and PSI-BLAST profiles as input ; together with the secondary structure, it also predicts solvent accessibility and coiled-coil regions.

Machine learning techniques have been taking an increasingly bigger role in the prediction of secondary structure: many of the aforementioned methods implement neural networks, and other techniques such as Support Vector Machines (SVMs) have been succesfully utilized for secondary structure prediction (Hua *et al*, 2001 [11]).

Between the methods presented in this brief introduction and all the other ones available, it is still not clear whether there is a method which is objectively better than the others; it is therefore important to compare them in performance and understand the differences between them.

The goal of this project is to compare the performance of the GOR and SVM methods: to do so, after training them with sequence profiles obtained from multiple sequence alignments, they are tested by predicting the secondary structure of a "blind set" of 150 sequences. The results show that the SVM method has a higher accuracy overall, possibly due to its capability to wholly handle the large feature space that is each sequence profile window, while the GOR method has to compromise and assume statistical independence for each element of the window.

2 Material and methods

2.1 Training dataset

The training of the models was performed on the dataset used in the training of the Jpred4 predictor, obtained from 1987 representative sequences from each superfamily in SCOP v2.04 (Fox *et al*, 2014 [12] : the choice of this heterogeneous dataset assures that all possible three-dimensional folds are covered, and reduces the likelihood of trivially detectable similarities between sequences; with this data, a good snapshot of the real protein space is provided.

From the original set of sequences, additional filtering steps were performed and 489 sequences removed:

- 3D structure with resolution lower than 2.5Å [306]
- sequence length lower than 30 residues and higher than 800 [27]
- domains made up of multiple chains [20]
- missing or incomplete DSSP information [110]
- inconsistencies between PDB, DSSP and other file definitions [17]

The resulting dataset was split into training [1348] and blind testing sets [150]. While the training set was used for the training of the models in this project, the blind testing set obtained from the Jpred dataset was not used, and a new one was generated from scratch. For each sequence of the training set, the residue sequence in FASTA format and the secondary structure sequence in DSSP format were provided.

2.1.1 Statistical analysis of the dataset

Some basic statistical analyses were performed on the dataset, in order to verify that it is indeed a good representative of the protein space; the results were compared with the UniprotKB/Swiss-Prot statistics as of December 11, 2019 (https://www.uniprot.org/statistics/Swiss-Prot).

1)Distribution of secondary structure conformations:

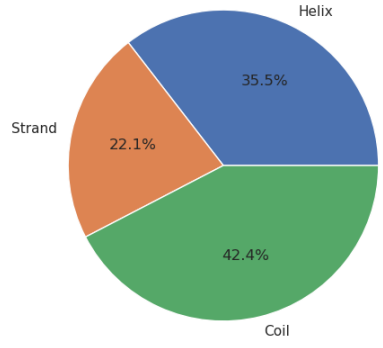

Figure 1: Distribution of secondary structure conformations

Figure 1 shows a relatively balanced proportion: all three categories are adequately present. The proportion of helix and strand conformations reflects the proportions found on Swiss-Prot; however, a reliable number for coil conformations on Swiss-Prot could not be determined, since on it only turn (T) conformations are considered, ignoring bend (S) and empty ("") dssp assignments. The number of turn conformations on Swiss-Prot is much lower than that of coil conformations in the training set; this apparent discrepancy could lead to some bias in the models, which would be more inclined to predict the more frequent structure.

2)Residue composition:

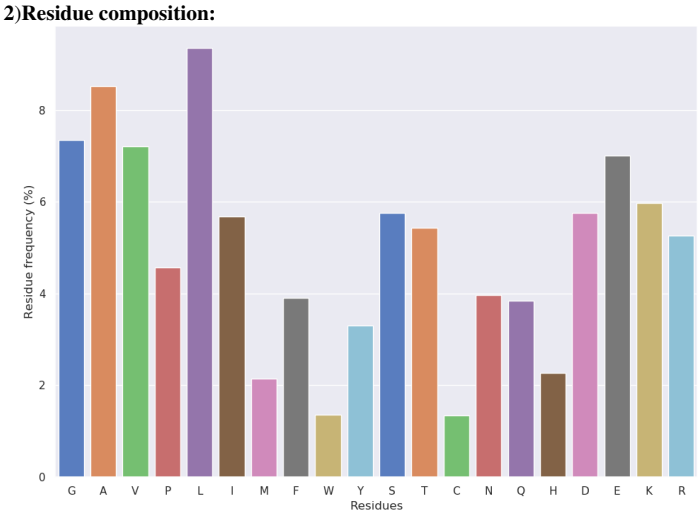

Figure 2: Residue composition of the dataset

The distribution of residues closely resembles the one obtained from Swiss-Prot.

3)Fraction of secondary structure conformations:

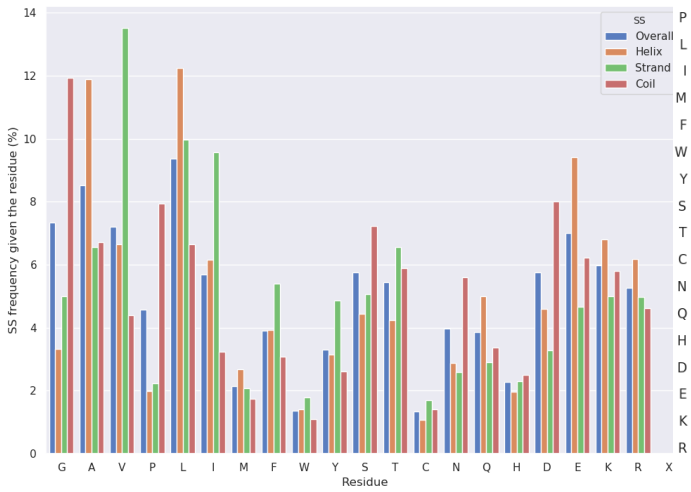

Figure 3: Distribution of secondary structure conformations given the residues

Amino acids have different atomic compositions and stereochemical properties, which means that they favour different secondary structure conformations. Proline(P) and glycine(G) residues disrupt the regularity of the α-helical backbone conformation, and are commonly found in turns. Amino acids that prefer to adopt helical conformations include leucine(L), alanine(A), glutamate(Q), methionine(M) and lysine (K); the large aromatic residues (tryptophan(W), tyrosine(Y) and phenylalanine(F)), along with isoleucine(I), valine(V), and threonine(T) mostly adopt β-strand conformations. As can be observed in figure 3, the training set doesn't deviate from these tendencies.

4)Composition of windows of 17 residues given the secondary structure of the central (eigth) residue:

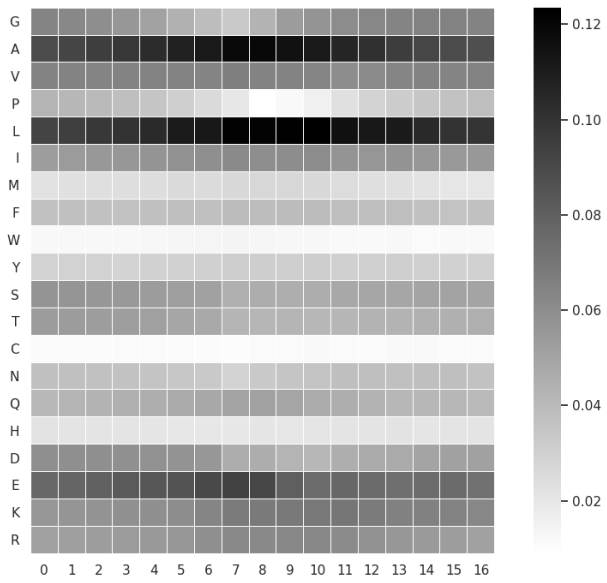

Figure 4A: Heatmap for the helix conformation; the relative frequency of each residue (vertical axis) in each position (horizontal axis) is represented by the color, ranging from white to black according to the scale on the side of the graph.

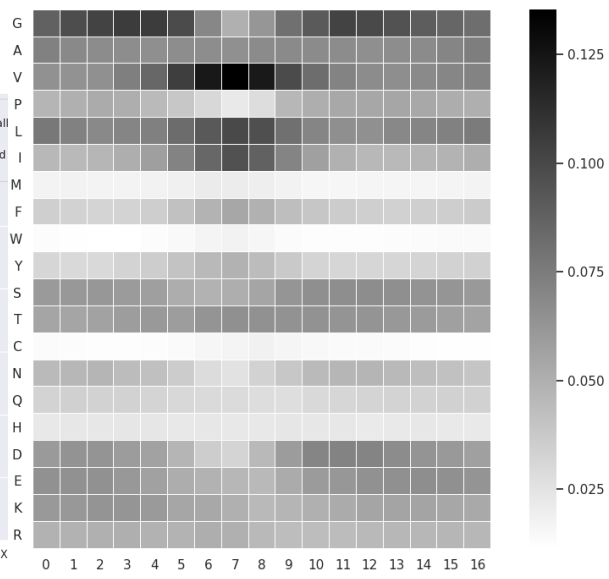

Figure 4B: Heatmap for the strand conformation

The most frequent residues around the central one in helix conformation are leucine, alanine and aspartate, as can be expected by their high propensity to form helices; the opposite is true for the proline and glycine residues, which either break the helix due to the steric hindrance and the inability to donate an amide hydrogen bond (proline) or disrupt its stability with its high conformational flexibility (glycine).
The strand conformation presents a higher concentration of valine residues around the central one, which is consistent with the tendency of the residue to appear in β-strands; as with helices, the proline frequency is higher when far from the central residue, since proline is often found at the edges of the strands to avoid unwanted associations with other edges from other proteins. The same goes for glycin, the frequency of which drastically increases when moving away from the central residue.

5)taxonomic classification

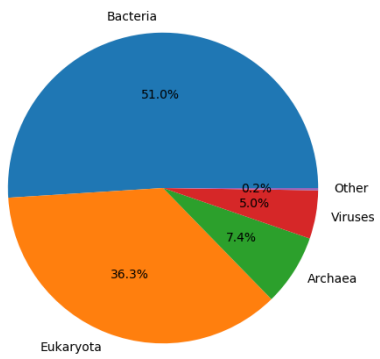

Figure 5: taxonomic composition of the dataset by kingdom

The taxonomic composition by species is available in the supplementary materials.

6)structural classification by SCOP [12] protein fold class

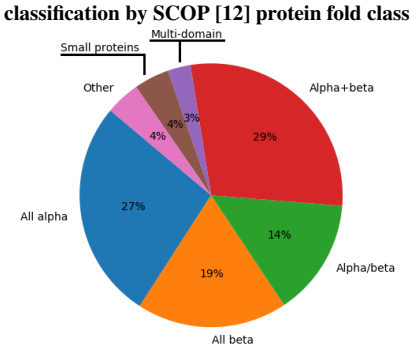

Figure 6

The same analyses were performed on the blind test, and all the distributions appear comparable to those of the training set; the resulting figures are available in the supplementary materials. Compared to the Swiss-Prot database, the training set appears to be a good representative overall: most of the sequences are of comparable length according to the distribution shown on the website, since they were filtered to be between 30 and 800 residues long; the distributions of residue and secondary structure frequency are very similar, and so are the distributions of kingdom (Figure 5) and SCOP class (Figure 6). The dataset covers all the main SCOP classes.

2.2 Blind testing dataset

Since the training of the model is performed on the whole Jpred4 dataset, in order to evaluate its final performance an additional dataset is required; the so-called "blind" set must be completely independent from the training set, in order to avoid biased predictions; the 150 sequences were selected to reproduce a similar distribution of secondary structure compositions as the training set, in order to avoid biasing the reported accuracy of the blind test results.

2.2.1 Selection of proteins and internal redundancy reduction

To obtain the set, a search on PDB [1] was performed with the following parameters:

- deposit date after January 2015 (after the release of the Jpred4 dataset in 2014)
- x-ray resolution under 2.5Å
- chain length between 50 and 300 residues
- wild type protein (no engineered mutations)
- retrieve only representatives at 30% sequence identity (to avoid redundancy)

An additional step of internal redundancy reduction was performed with blastclust (Dondoshansky $et\,al$, 2000 [13]) : the sequences obtained from PDB having a similarity higher than 30% over an area covering more than 30% of their length were clustered together, and a representative of each of the 3113 clusters obtained was selected.

2.2.2 External redundancy reduction

However, there may still be some similarity with elements of the training set: to avoid this redundancy, the cluster representatives were formatted into a BLAST database with the makeblastdb tool from the BLAST+ suite of programs (Camacho $et\,al$, 2009 [14]). The training set was then compared to the database with blastp([14]): all sequences with similarity higher than 30% were removed from the list.
From the list, 150 sequences were randomly selected to build the final blind set.

2.2.3 Assigning secondary structure to the sequences

From the PDB files of the blind set, the secondary structure sequence was inferred with DSSP (Define Secondary Structure of Proteins, Kabsch $et\,al$, 1983 [15]). DSSP implements an algorithm which calculates the most probable secondary structure assignments given the geometry of the protein's 3D structure. The main idea behind the program is that the different secondary structure types in the atomic coordinates of a protein can be discriminated by identifying the corresponding hydrogen bonding patterns. Instead of using backbone ϕ and ψ angles or Cα positions, which require the adjustment of many parameters, the presence or absence of an H-bond can be characterized by a single parameter, which is the cutoff in the bond energy. Structures obtained with X-ray cristallography do not contain most hydrogen atom coordinates, since the wavelength of X-rays (circa 1Å) is too large to detect them. DSSP discards any present hydrogen atom from the PDB file and places a hydrogen atom at 1Å from each backbone N atom, in the opposite direction from the backbone C=O bond. The electrostatic energy of each potential H-bond between all atoms is then calculated, and if it is lower than 0.5 kcal/mol then the bond is considered existent. DSSP defines eight types of secondary structure according to their H-bonding patterns; for the purpose of this project, they were grouped as following:

- 3,4, and 5-turn helix (G,H,I)=> **Helix (H)**
- residue in isolated β-bridge (B) and extended strand participating in β-ladder (E) => **Strand (E)**
- H-bonded turn (T), bend (S) and no assignment("", empty) => **Coil (C or -)**

Finally, the secondary structure sequences were extracted from the dssp output files, along with the corresponding primary structure.

2.3 Obtaining the sequence profiles

As introduced earlier, using evolutionary information when training secondary structure predictors yields better results. Multiple sequence alignments of sequences belonging to proteins of the same family contain much more information than the single sequences, since conserved regions usually indicate structurally or functionally important residues.
Sequence profiles are a way to represent in a more compact way multiple sequence alignments: a matrix of dimension L x 20, where L is the length of the target protein sequence and 20 the number of possible residues. Each element of the matrix represents the occurrence frequency of each residue on each aligned position. For the purpose of this project, the models were trained and tested with sequence profiles rather than single sequences; in order to obtained the profiles, each sequence of both training set and blind testing set was searched by similarity against a large sequence database. To obtain the database, all the sequences present on UniprotKB/SwissProt were indexed with the makeblastdb tool, specifying the protein nature of the sequences.
PSI-BLAST (Position-Specific Iterative BLAST, Altschul $et\,al$, 1997 [16]) was then chosen to search for homologous sequences in the database. While BLAST runs on a single iteration, PSI-BLAST performs multiple search iterations using a Position Specific Substitution Matrix (PSSM) computed during the search. On the first iteration, a standard BLAST search is performed using a pre-defined substitution matrix, such as BLOSUM62; a multiple sequence alignment is computed by stacking the pairwise local alignments obtained; the PSSM is then computed from the MSA. In the following iterations, BLAST searches are performed using the PSSM as substitution matrix, and multiple sequence alignments are computed from the sequences found using the PSSM obtained in the first iteration;the aligned sequences are clustered by sequence identity. Finally, after a user-defined number of iterations, or if no more alignments can be found with e-value below the specified treshold, the algorithm stops and

the MSA is returned along with the PSSM.
To obtain the profiles needed in the project, each sequence underwent three iterations of PSI-BLAST run against the database with an e-value threshold of 0.01 and number of alignments set to 1000, due to the large size of the database. Some target sequences returned empty profiles (due to PSI-BLAST not finding any similarity within the set parameters) or did not present a PSI-BLAST output; they were removed from the training set, leaving a number of 1199 sequences for the training set, while the blind set was built only with sequences that returned non-empty profiles.

2.4 The GOR method

2.4.1 Description

The Garnier-Osguthorpe-Robson (GOR) method was originally developed in 1978 [5], and implements information theory and bayesian statistics for protein secondary structure prediction. Over the years, it underwent several improvements and refinements, with the addition of larger databases, the detailing of statistics accounting for amino acid pairs and triplets (Gibrat $et\,al$, 1987 [17]), and most importantly the inclusion of evolutionary information in the form of sequence profiles (Kloczkowski $et\,al$,2002[18]). The underlying idea is similar to some previous methods, such as Chou-Fasman: the observed frequency of residues in a particular secondary structure conformation can help predict the secondary structure starting only from the primary structure. The extent to which the presence of a particular residue R influences the probability of having the conformation S is determined with the following information function:

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

where P(S|R) is the conditional probability of observing the conformation S when the residue is R, and P(S) is the marginal probability of observing the conformation S. Using the probability chain rule, the equation can be rewritten as

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

Where P(R,S) is the joint probability of observing residue R in conformation S, and P(R) is the marginal probability of observing the residue R. The novelty of the GOR method lies in the consideration of the influence of neighbouring residues on the conformation of the given residue. This is reflected in the information function, which is extended over a d number of residues preceding and following the central one:

$$I(S;R_{-d},...R_0,...R_d) = log\frac{P(S|R_{-d},...R_0,...R_d)}{P(S)} = \tag{3}$$

$$= log\frac{P(S,R_{-d},...R_0,...R_d)}{P(R_{-d},...R_0,...R_d)P(S)} \tag{4}$$

The value of d is usually 8; this number is not arbitrary but was based on studies of information content at increasing separations [19]. However, calculating the joint probabilities of S and all the possible residues in all the possible positions in the window isn't computationally feasible, and would require databases much larger than currently available to obtain reliable probabilities. To avoid this problem, the residues in the window are considered to be statistically independent. The formula then becomes :

$$log\frac{P(S)\prod_{k=-d}^{d}P(R_k|S)}{P(R_{-d},...R_0,...R_d)P(S)} = log\prod_{k=-d}^{d}\frac{P(R_k,S)}{P(S)P(R_k)} \tag{5}$$

$$= \sum_{k=-d}^{d} I(S;R_k) \tag{6}$$

Each $I(S;R_k)$ of the summatory can be computed as already seen in equation 2. Each parameter in the mentioned equations is derived from the data used in the training of the model:

- $P(R_k,S)$: frequency of residues of type R observed at position k in windows where the central residue R_0 is in conformation S
- $P(R_k)$: frequency of observed residues at position k
- $P(S)$: frequency of each conformation S

In the more recent iterations of the GOR method, the input is in the form of sequence profiles: the information function is then represented as

$$I(S;PW) = \sum_{k=-d}^{d}\sum_{R_k} PW[R_k] * I(S;R_k) \tag{7}$$

Where $I(S;R_k)$ is calculated using the residue frequencies from the matrix obtained from the training set (see implementation section), and then multiplicated by the corresponding residue frequency in the profile window of the sequence to predict.
The highest value of the three possible information functions (H,E,C) determines the most likely conformation for the central position of the considered window in the sequence profile.

2.4.2 Implementation

The implementation of the GOR method was done using two python scripts developed in-house: one for the training of the model, and one for the prediction of the testing set. The input data for the training consists of the sequence profiles and the corresponding dssp files containing the secondary structure of the sequences which the profiles were based on; the output is four 17x20 matrices, each containing the window of residue frequencies for each secondary structure conformation of the eighth residue position, and a window of overall frequencies of residues. The prediction script simply computes the information function shown in equation 7 by using the matrices obtained in the training and the sequence profile of the input protein chain. The scripts are available in the supplementary materials.

2.5 Support Vector Machines

2.5.1 Description

Support Vector Machines (SVM) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis (https://en.wikipedia.org/wiki/Support-vector_machine). The first implementation of SVMs for secondary structure prediction goes back to 2001, when S.Hua and Z.Sun assembled a tertiary classifier for the three conformations H,E, and C from the combination of several binary classifiers [11]. The method achieved notable results, with a three-state accuracy of 73.5% Compared to other machine learning techniques such as neural networks, SVMs have the advantage of being less prone to overfitting and being able to handle large feature spaces; the implementation of the kernel function allows the implicit mapping of the data to a high-dimensional space, which facilitates the separation of the data in classes. In this project, SVMs were used only for classification, and not for regression.

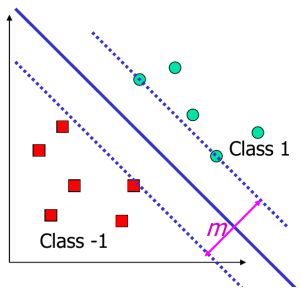

Fig. 1: Hyperplane separating the feature space in two classes (from Pier Luigi Martelli - Systems and In Silico Biology 2019-2020- University of Bologna)

The way SVMs work is by calculating the hyperplane which divides the feature space in two classes, while maximizing the distance between the support vectors (margin, m in fig. 1), which are defined as the closest points for each class to the hyperplane.
Given a training set with n elements, each having a value $x_i \in \mathbb{R}^d$ and belonging to a class $y_i \in \{-1, +1\}$, we define the hyperplane with the equation $w^T x + b = 0$. We can then say that $y_i(w^T x_i + b) \geq \frac{\rho}{2}$, with ρ being the margin. The support vectors are the points where $y_s(w^T x_s + b) = \frac{\rho}{2}$; after rescaling w and b by $\frac{\rho}{2}$ the distance from them to the hyperplane is:

$$r = \frac{y_s(w^T x_s + b)}{\|w\|} = \frac{1}{\|w\|} \tag{8}$$

And therefore ρ can be expressed as $2r = \frac{2}{\|w\|}$
The maximization of ρ then becomes the following optimization problem:

$$minimize \quad \frac{1}{2}\left\|w^2\right\|, \ subject\,to \quad y_i(w^T x_i + b) \geq 1 \, \forall i \tag{9}$$

To solve it, a Lagrange multiplier αi is introduced to construct the dual problem:

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

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

A global maximum of αi can always be found; w can be recovered with $w = \sum_{i=1}^{n} \alpha_i y_i x_i$, even though it is not explicitly needed for the classification of new points. All x_i with non-zero αi are support vectors. b can be obtained with $b = y_k - \sum_s \alpha_s y_s x_s^T x_k$. The classifying function for new points becomes:

$$y(x) = \sum_s \alpha_s y_s (x_s^T x) + b \tag{11}$$

With x being the new point. This formulation however admits only training sets with all points grouped in a way in which they are linearly separable: in the case of even a single outsider, the model must be able to consider it as an error up to a certain extent. To do so, the so called "soft margin" is used: a slack variable ξi, being the upper bound of allowed errors, is introduced in the optimization problem (equation 9) in the following way:

$$minimize \quad \frac{1}{2}\left\|w^2\right\| + C\sum_{i=1}^{n} \xi_i \tag{12}$$

$$subject\,to \quad y_i(w^T x_i + b) \geq 1 - \xi_i \quad with\, \xi_i \geq 0, \ \forall i$$

C being the hyperparameter that weighs the maximization of the margin, even if it includes errors, against the minimization of errors on the training points.
In some cases the training set is not linearly separable, even with the implementation of soft margins. The original input space however can always be transformed to a new feature space (usually higher dimensional), where linear operations correspond to non-linear operations in the original space; a good transformation can render linearly separable problems which are not linearly separable in their original dimension. As seen in equation 11, the classification relies on the scalar product between support vectors and the unlabeled point, which after a transformation ϕ becomes $K(x_s, x') = \phi(x_s^T)\phi(x')$. This is the kernel function, which implicitly remaps the data without explicitly carrying out the transformation of all the input data. The classifying function can be updated by substituting the scalar product with the kernel function. There are several kernel functions that can be used with support vector machines; the one utilized in this project is the gaussian/radial basis function (RBF):

$$K(x_s, x') = exp(-\gamma \left\|x_s - x'\right\|^2) \tag{13}$$

With $\gamma = \frac{1}{2\sigma^2}$; the input space is transformed to an infinite-dimensional feature space, where every point is mapped to a gaussian function. The combination of the functions for the support vectors determines the separator.

2.5.2 Implementation

The training and testing of the SVM models were performed using the LIBSVM library, version 3.24 (Chang *et al*, 2011 [20]). LIBSVM is one of the most popular SVM implementations, and supports both regression and multiclass classification (in the form of combined binary classifications), with several kernel functions available; other parameters such as the value of gamma (γ) in the gaussian kernel function, or the cost parameter C for the soft margin optimization can be adjusted. The input data for implementing classification with LIBSVM is required to be in the following format:
<class> <feature_index>:<feature_value>
To produce the input data, each sequence profile obtained in step 2.3 was associated with the DSSP file corresponding to the target sequence; for each window of 17 residues, the secondary structure of the central residue was registered as the class, while each element of the 17x20 profile window was registered as the feature. The procedure was iterated over the whole profile, obtaining a file of containing one input vector for each residue of the sequence, with the $class$ attribute being 1,2 or 3 depending on the secondary structure (1 for H, 2 for E, 3 for C), and the feature space constituted of at most 340 features : one for each frequency found in the profile window, omitting all features with value 0. The training was then performed four times on each set, changing on each iteration the γ parameter (-g) and the cost parameter C (-c), in order to find the parameters which yield a better model:

- -g 0.5 , -c 2
- -g 0.5 , -c 4
- -g 2 , -c 2
- -g 2 , -c 4

2.6 Scoring indexes

As explained in the preceding sections, secondary structure prediction, as performed in this project, is a classification problem with multiple classes: each residue can be part of three possible conformations, and the two methods implemented slightly differ in their approach. While the GOR method computes an information function for each possible conformation, SVM trains three binary classifiers which discriminate

between two possible conformations each (H/E, H/C, E/C) by computing a discrimination function for each classifier; in both methods however, the class is predicted by comparing the value of the three functions obtained: the one with the highest value will determine the class. The evaluation of the model performances can be performed by using the same scoring indexes, since the output is comparable.

2.6.1 Multi-class confusion matrix

First, a three-class confusion matrix is computed:

		Observed		
		H	E	C
Predicted	H	p_{HH}	p_{HE}	p_{HC}
	E	p_{EH}	p_{EE}	p_{EC}
	C	p_{CH}	p_{CE}	p_{CC}

Table 1. Three-class confusion matrix; the columns indicate the observed classes, while the rows indicate the predicted classes. p_{XY} indicates the object in class Y predicted to be in class X

The three-class accuracy Q_3 is calculated by dividing all the correct predictions by the total number of predictions:

$$Q_3 = \frac{p_{HH} + p_{EE} + p_{CC}}{N} \tag{14}$$

2.6.2 Binary scoring indexes

From the 3-class confusion matrix a binary class matrix is extracted for each secondary structure conformation:

		Observed	
		E	Not E
Predicted	E	c_E	o_E
	Not E	u_E	n_E

Table 2. Binary scoring matrix

- c : correct positive predictions
- o : overpredictions
- u : underpredictions
- n: correct negative predictions

The following scoring indexes are evaluated:

- $SEN_{SS} = \frac{c_{SS}}{c_{SS}+u_{SS}}$

  Sensitivity or true positive rate : measures the proportion of actual positives that are correctly identified as such

- $PPV_{SS} = \frac{C_{SS}}{c_{SS}+o_{SS}}$

  Precision or positive predictive value : proportion of observed positive values over all the positive predictions

- $MCC_{SS} = \frac{(c_H * n_H) - (o_H * u_H)}{\sqrt{(c_H+o_H)*(c_H+u_H)*(n_H+o_H)*(n_H+u_H)}}$

  Matthew's correlation coefficient: measures the correlation between observed and predicted binary classifications, taking into account all four quadrants of the binary matrix. While the other scoring indexes have values that go from 0 to 100%, the MCC goes from -1 to +1.

2.6.3 Segment overlap index

While confusion matrices and the deriving scoring indexes are useful to evaluate the performance of the model and verify that it works as intended, they do not take into account the need to obtain biologically relevant predictions. The evaluation must consider more than just the percentage of overlapping individual positions between the observed and predicted sequence, otherwise a seemingly well-performing predictor may lead to a three-dimensional structure not at all corresponding with the original structure. For example, predicted α-helices shorter than four residues are biologically meaningless, since the bond that characterizes them is between residues located at a distance of around four residues in the sequence; in the same way, β-strands usually involve two or more residues. The implementation of the GOR and SVM models presented in this project is a rather crude one, since these factors are not taken into account; to mitigate this problem, the evaluation of the model performance also considers the overlapping of sequence segments, which is useful in understanding the biological correctness of predictions.

The Segment OVerlap index (SOV), first introduced by Rost et al. in 1994 [21], aims to capture the biological significance of secondary structure predictions, by considering the overlapping segments rather than the per-residue assignments of conformation. In this project, a modified and updated definition of SOV was utilized (Zemla *et al*, 1999 [22]): a correction (δ) was added to the original formula to define a degree of variation allowed at the segment edges, so as to deal with uncertainty in the experimental segment boundary assignment, and made simmetric with respect to observed and predicted segments; additionally, the normalization value is updated to reflect the pairwise nature of segment comparison: the prediction score for erroneous partitioning and non-prediction of segments is lowered, and the score becomes a range from 0 to 100%. The updated formula is then presented as follows:

$$SOV(S) = 100 * \frac{1}{N_S} \sum_{\{(S_o,S_p)|S_o \cap S_p \neq \varnothing\}} [\frac{minov(S_o,S_p) + \delta(S_o,S_p)}{maxov(S_o,S_p)} * len(S_o)] \tag{15}$$

With:

- S_o and S_p observed (o) and predicted (p) segments
- N_s number of residues in conformation S (normalization value)
- $(S_o, S_p)|S_o \cap S_p \neq \varnothing$ set of observed and predicted segments overlapping by at least one residue
- $minov(S_o, S_p) = S_o \cap S_p$ length of the overlap of S_o and S_p
- $maxov(S_o, S_p) = len(S_o + len(S_p - minov(S_o, S_p)))$ length of the total extent of S_o and S_p
- $\delta(S_o, S_p)$ = min$\{maxov(S_o, S_p\}$, $minov(S_o, S_p)$, $len(S_o/2)$, $len(S_p)/2\}$

2.7 Cross evaluation

Before undergoing the final testing on the blind set, it must be verified that the models have a consistent performance, since accuracy can vary significantly depending on which set of proteins is chosen as the testing set. To do so, the scoring indexes were averaged over independent trials with five distinct partitions of the 1199 training sequences into training and testing set (5-fold cross-validation); each training set comprising 4/5ths of the sequences and the testing set the remaining 1/5th. The standard error was then computed for each averaged scoring index (except for SOV), with the following formula:

$$SE = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}}{\sqrt{n}} = \frac{\sigma}{\sqrt{n}} \tag{16}$$

Where n is the number of tested partitions, $\bar{x}$ is the mean of the n values and σ is the standard deviation.

In order to be biologically meaningful, the SOV index is computed over each whole protein chain rather than on the scoring matrix; therefore, its average and standard error must be computed over the whole set of protein chains predicted in the five tests rather than over the five partitions.

3 Results

All the SVM models obtained with different options for γ and C, as described in section 2.5.2, were compared, and the values of $\gamma = 0.5$ and $C = 2$ returned the best cross-validated performance over all scoring indexes; therefore, only the models obtained with these parameter values are considered for the comparison with GOR. All the performance data of the discarded models is available in the supplementary materials.

CV	SS	GOR	SVM
SEN	**H**	80.07 ± 0.45	68.93 ± 0.92
	E	70.87 ± 0.67	40.62 ± 1.02
	C	43.37 ± 0.38	87.83 ± 0.3
PPV	**H**	63.41 ± 0.59	84.79 ± 0.39
	E	48.55 ± 0.9	78.92 ± 0.74
	C	80.11 ± 0.38	62.3 ± 0.39
MCC	**H**	0.53 ± 0.004	0.66 ± 0.007
	E	0.44 ± 0.004	0.49 ± 0.006
	C	0.42 ± 0.003	0.49 ± 0.003
SOV	**H**	74.86 ± 0.68	74.75 ± 0.94
	E	71.58 ± 0.74	52.13 ± 1.08
	C	51.38 ± 0.51	66.85 ± 0.74
Q3		62.46 ± 0.08	70.67 ± 0.27

Table 3. Cross-validation results of the GOR and SVM models; each averaged scoring index value is accompanied by its standard error; the SOV index is averaged over the total number of sequences predicted. The first column contains the scoring indexes as abbreviated in section 2.6.2, while the second column divides the indexes by the secondary structure conformation on which they were computed.

The results of the cross-validation reveal no significant difference in performance between the different partitions; no standard error is higher than 1.02%, which confirms the overall quality of the training set; a small difference in performance between the partitions is to be expected, since they do not contain the exact same number and distribution of residues and associated secondary structure conformations.

As can be seen in table 3, SVM seems to perform better overall: only SEN_E, PPV_C and SOV_E have significantly lower values when compared with the GOR model; the three-class accuracy (Q_3) of SVM is significantly higher, and so is the MCC.

The performance of the models on the blind testing shows very similar values to the cross-validation results (Table 4). The standard error of the SOV index is higher when compared to the tests on the training set, which is understandable due to the significantly smaller size of the blind testing set, which makes it more impacted by fluctuations (150 sequences compared to the 1199 sequences predicted over 5-fold cross-validation of the training set).

BS	SS	GOR	SVM
SEN	**H**	75.65	64.4
	E	69.36	45.8
	C	46.72	89.68
PPV	**H**	66.27	87.74
	E	51.86	84.64
	C	73.12	57.21
MCC	**H**	0.50	0.63
	E	0.45	0.55
	C	0.41	0.49
SOV	**H**	67.86 ± 1.97	68.28 ± 2.7
	E	74.6 ± 1.80	57.87 ± 2.9
	C	51.03 ± 1.3	64.51 ± 1.95
Q3		63.35	69.45

Table 4. Performance of the GOR and SVM models on the blind testing set

4 Discussion

The statistical analyses and the results of the cross-validation on the training set confirm it as a good representative of the protein space, with low internal redundancy and high enough variation to cover all the protein folds, residues and secondary structure conformations.

The influence of the local residue context on the secondary structure conformation of the single residue has been thoroughly demonstrated; while the training of the SVM was able to take into account the full correlation between neighbouring residues and the conformation of the central residue, thanks to its ability to handle large feature spaces with kernel functions, the GOR method is based on the assumption of statistical independence between the residues in the window; this simplification may be one of the reasons why it underperforms when compared to SVM.

Other than the inclusion of evolutionary information in the form of sequence profiles and the usage of a rather large training dataset, the implementation of GOR shown in this project didn't employ many of the improvements that the method underwent over the years, such as the consideration of amino acid pairs or triplets.

As already anticipated in section 2.6.3, both SVM and GOR implementations didn't consider some structural properties of secondary structure that give biological meaning to the predictions, such as the need for at least four residues in H conformation to form an helix, or two in E conformation to form a strand-like structure; the SOV index and consequently the biological correctness of the predictions would certainly benefit from the implementations of such constraints in the models.

Despite these limitations, both models show promising results; the support vector machine approach in particular could provide a good predictor if these limitations were to be addressed.

5 Conclusion

In this project, two methods for secondary structure prediction using evolutionary information in the form of sequence profiles were presented, implemented and tested. The GOR method, using Bayesian statistics and information theory concepts, and the support vector machine (SVM) method, a machine learning model for multi-class classification through the combination of binary classifiers. Both methods were able to predict the secondary structure of a set of protein chains starting from their sequence profiles with a three-class accuracy (Q_3) higher than 60%; the SVM method was able to reach 70%. Some possible improvements were presented, such as taking into account the structural constraints that concern segments of secondary structure. In the end, the SVM method appears to be the one more fit to be implemented in a framework for the prediction of secondary structure and functional annotation.

References

[1]Berman et al. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 01 2000.

[2]The UniProt Consortium. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research*, 47(D1):D506–D515, 11 2018.

[3]V.I. Lim. Structural principles of the globular organization of protein chains. a stereochemical theory of globular protein secondary structure. *Journal of Molecular Biology*, 88(4):857 – 872, 1974.

[4]Peter Y Chou and Gerald D Fasman. Prediction of protein conformation. *Biochemistry*, 13(2):222–245, 1974.

[5]Jean Garnier, David J Osguthorpe, and Barry Robson. 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, 1978.

[6]Steven A Benner and Dietlinde Gerloff. Patterns of divergence in homologous proteins as indicators of secondary and tertiary structure: a prediction of the structure of the catalytic domain of protein kinases. *Advances in enzyme regulation*, 31:121–181, 1991.

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

[8]Christophe N Magnan and Pierre Baldi. Sspro/accpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. *Bioinformatics*, 30(18):2592–2597, 2014.

[9]Alexey Drozdetskiy, Christian Cole, James Procter, and Geoffrey J. Barton. JPred4: a protein secondary structure prediction server. *Nucleic Acids Research*, 43(W1):W389–W394, 04 2015.

[10]James A Cuff and Geoffrey J Barton. Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 40(3):502–511, 2000.

[11]Sujun Hua and Zhirong Sun. A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *Journal of molecular biology*, 308(2):397–407, 2001.

[12]K Fox Naomi and E Brenner Steven. Chandonia john-marc. scope: Structural classification of proteins–extended, integrating scop and astral data and classification of new structures. *Nucleic Acids Res*, 42(1):D304–D309, 2014.

[13]I Dondoshansky and Y Wolf. Blastclust, 2000.

[14]Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L Madden. Blast+: architecture and applications. *BMC bioinformatics*, 10(1):421, 2009.

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

[16]Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402, 1997.

[17]J-F Gibrat, J Garnier, and Barry Robson. Further developments of protein secondary structure prediction using information theory: new parameters and consideration of residue pairs. *Journal of molecular biology*, 198(3):425–443, 1987.

[18]A Kloczkowski, K-L Ting, RL Jernigan, and J Garnier. Combining the gor v algorithm with evolutionary information for protein secondary structure prediction from amino acid sequence. *Proteins: Structure, Function, and Bioinformatics*, 49(2):154–166, 2002.

[19]Jean Garnier, Jean-François Gibrat, and Barry Robson. [32] gor method for predicting protein secondary structure from amino acid sequence. In *Methods in enzymology*, volume 266, pages 540–553. Elsevier, 1996.

[20]Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[21]Burkhard Rost, Chris Sander, and Reinhard Schneider. Redefining the goals of protein secondary structure prediction. *Journal of molecular biology*, 235(1):13–26, 1994.

[22]Adam Zemla, Česlovas Venclovas, Krzysztof Fidelis, and Burkhard Rost. A modified definition of sov, a segment-based measure for protein secondary structure prediction assessment. *Proteins: Structure, Function, and Bioinformatics*, 34(2):220–223, 1999.