*Bioinformatics*, YYYY, 0–0

doi: 10.1093/bioinformatics/xxxxx

Advance Access Publication Date: DD Month YYYY

Manuscript Category

|  |
| --- |
| Comparing the GOR method with Support Vector Machines for the prediction of protein secondary structure  Asole Giovanni1  1Department of Pharmacy and Biotechnology, MSc in Bioinformatics.  Abstract  **Motivation:** Obtaining a good prediction of the secondary structure of a protein is one of the most important topics in the field of Bioinformatic, as important information can be obtained from them, including the tertiary structure, functions, interactions with other proteins, subcellular locations and much more. Therefore, we compared two machine learning based methods for secondary structure prediction starting from the primary sequence: Garnier-Osguthorpe-Robson (GOR) and a Support Vector Machine (SVM).  The two methods were learned on the entire jPRED4 dataset and after that their ability to predict and generalize was tested on a blind-set.  **Results:** Both SVM and GOR method achieved a good secondary structure prediction. The Support Vector machine method surpassed  the GOR with a three class accuracy respectively of 0.773 and 0.610.  **Contact:** Giovanni.asole@studio.unibo.it  **Supplementary information:** Supplementary data are available at <https://github.com/asolee/lb2-2021-project-asole>. |

# Introduction

The only way to fully understand many characteristics of proteins including function, interactions, localization, is to know their tertiary structure. Unfortunately, obtaining the tertiary structure of a protein is not an easy task, nowadays experimental methods are widely used, among them the "standard method" is X-ray

crystallography (Bond, 2014), in comparison to other methods including Nuclear Magnetic Resonance (NMR) and Cryo-electron ​​microscopy. The use of these experimental methods, however, cannot be compared with the more than exponential growth of datasets seeking to provide high-quality information on protein function, such as UniProt. For this reason, one of the most important tasks of Bioinformatic is the prediction of the tertiary structure of a protein starting from its primary sequence, to do this it is necessary to have effective and very precise methods that allow to obtain the secondary structure starting from the primary sequence.

Protein secondary structure prediction began in 1951 when Pauling and Corey predicted helical and sheet conformations for protein polypeptide backbone even before the first protein structure was determined. Secondary structure prediction techniques have been classified into three generations [1].

In the first generation, secondary structures were predicted from a protein sequence according to statistical propensities of aminoacid residues towards a specific secondary structure element, The most representative method of first-generation methods is the Chou–Fasman method [2]. The second-generation methods, represented

by the Garnier-Osguthorpe-Robson (GOR) method [3] and the Lim method [4], used a sliding window of neighbouring residues and various theoretical algorithms such as statistical information, as well as other methods that use a learning method including neural networks [5]. The information obtained from the neighboring residuals allowed to increase the precision of the prediction up to 60%.

The third generation of techniques is characterized by using evolutionary information derived from alignment of multiple homologous sequences [6]. During this period, new computational algorithms have been implemented. Examples are support vector machines [7], hidden Markov network [8], PSIPRED [9].

With the third generation the accuracy on the prediction of the secondary structure reaches 70-75%.

The work described in this publication is based on the use of two main methods for the prediction of the protein secondary structure: GOR and SVM. The dataset used to train the two methods is jPRED4 dataset composed of 1348 protein and a test set (blind-set) composed of 150 proteins.

For both methods, a cross-validation phase was performed first to fix the hyper parameters, followed by a train phase on the entire dataset to produce models. Finally, the two models obtained by the two different methods were tested on blind-tests obtaining a prediction of the structure, which was compared with that one produced using the DSSP program (reduced to the three main structures), in order to retrieve indices that allow us to compare the precision of the secondary structure prediction of the two methods.

This procedure finally showing that SVM method work better than the GOR one.

# Materials and Methods

## Training dataset description

The training dataset is the same used from by the authors of JPred4 to train their predictor [10].

The initial jpred dataset started with 1987 single representative sequences from each superfamily in SCOPe v.2.04.

The number of sequences has been reduced to 1497 by applying the following filters:

* Resolution > 2.5 Å.
* Sequence length > 30 and < 800 residues.
* Missing DSSP information for more than 9 consecutive residues.
* Accession mapping inconsistencies.
* failed to produce PSI-BLAST hits.

After wich the dataset has been split into train and test set, 1348 and 149 sequences, respectively. In these project we only considered the train set, instead the test set was generated separately.

# Training dataset statistical analysis

A basic statistical analysis was carried out to prove that the dataset under consideration has a good quality and a good distribution in term of characteristics.

The distribution of the domain length of our train dataset is shown in the figure below (Fig. 1).

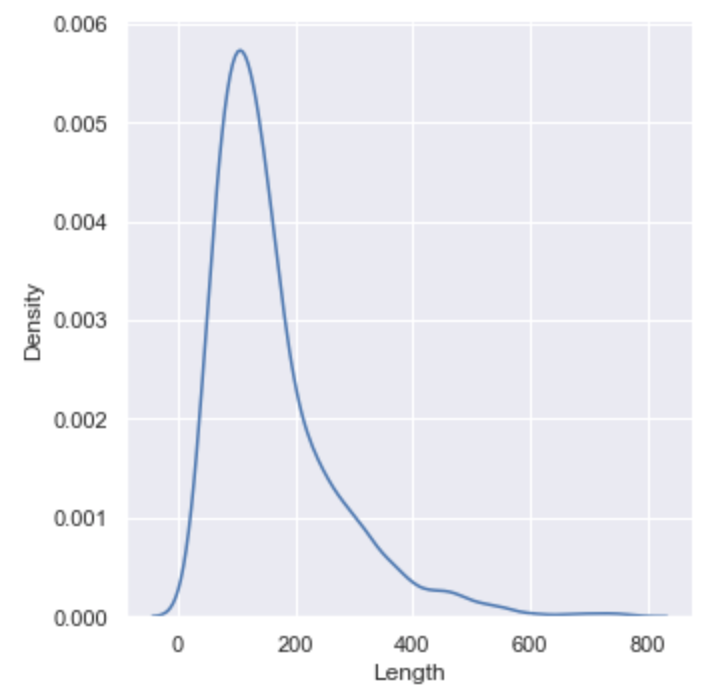


Figure 1: sequences length distribution in the train set (jpred4)

the greater density of distribution is observed around 150/160 residues.

The analysis of the SCOP class aboundance (Fig. 2) show that the classes are quite balanced, in which the most present class is alpha and beta (a + b), as well as the all alpha and all beta sequences represent a large portion of the dataset

Chart, pie chart

Description automatically generated

Figure 2: distribution of the SCOP classes in the train dataset (jpred4)

during the statistical analysis process on the train dataset, a control of the distribution in Super Kingdoms and Species was also carried out (Fig. 3, Fig. 4).

from the information obtained we can see that a large majority of the sequences fall within the bacterial classification, followed by eukaryotes.

Instead in terms of species it can be noted that the two major classes are Homo Sapiens and Escherichia Coli

Chart, pie chart

Description automatically generated

Figure 3: dataset distribution by Super Kingdoms

Chart, pie chart

Description automatically generated

Figure 4: dataset distribution by Species

From the graph shown (Fig. 5) can be seen that the length distribution along the Super Kingdoms is very similar and the distribution peak is almost always the same, this is because the train dataset is composed by representative sequences of the SCOP superfamilies that brings together more distantly related protein domains.

Chart, line chart

Description automatically generated

Figure 5: sequences length distribution based on Super Kingdoms

Instead, through a more in-depth analysis of the fasta and dssp files of the train set it was possible to analyze the distribution of the three main classes of the secondary structure (Fig. 6), as well as their distribution based on residues (Fig. 6).

## Blind test set description

After carrying out the cross-validation and the train procedure for both methods, a dataset is needed on which to test the precision of the secondary structure prediction, for this reason it was necessary to use an additional dataset called blind-test or holdout which is completely independent of the train set.

To generate the blind-set we started from an advanced search on PDB with the following filters:

* Deposit date after january 2015, to be sure that the selected structures are not part of jPred4 dataset (released in 2014)
* Chain length between 30 and 800 residues
* Only protein entities, no RNA/DNA
* Only X-ray diffraction experimental method,
* with a structure resolution greather than 2.5 Å.
* Only entities compatible with PDB format

After this filters, more than 21 000 sequences have been found.

To obtain a blind-set useful for the purpose, through the use of BALSTClust algorithm [11], the internal redundancy was reduced, in this process the sequences with a sequence identity greater than 30% and a coverage of at least 50% were clustered. The longest sequence was used as representative for each cluster.

On the sequences obtained in the previous step a reduction of the external redundancy was made, so that none of the sequences exceeded the 30% sequence idendity with our train set, otherwise the results obtained by our models would not have been fully reliable. The process takes two steps:

* The first step consists in creating a database based on the sequences chosen, through the use of makeblastdb from the BLAST+ [12].

This database will contain information that will allow you to continue with the next step

* In the second step, through the use of blasp [13], the sequences obtained with our train set were compared, and consequently we remove from the blind-set those with a sequence identity greater than 30%.

Finally the secondary structure in DSSP format [14] was assigned to the sequences of the blind-set and finally 150 random sequences were extracted to create the blind-test-set that will be used to evaluate the models.

1. **Blind test set statistical analysis**

To make sure that the blind test set is a good dataset to test the models, a statistical analysis was performed.

In particular, the composition of the secondary structure (Fig 6) and their distribution based on residues (Fig 6) were analyzed.

In this way it was possible to compare the results obtained from the train set and the blind test set.

Chart, bar chart, box and whisker chart

Description automatically generated

Figure 6: secondary structure distribution given the residue and general secondary structure conformation for both train and blind test set.

## Sequence profile generation

By comparing several similar sequences belonging to the same family (possibly distantly related homologs), it is possible to construct a sequence profile, i.e. a matrix composed of as many rows as the length of the sequence and of twenty columns that represent the residue, in each position of the matrix the probabilities of finding the residue in that specific spot of the sequence will be saved.

The generation of a sequence profile allows to obtain much more detailed information on the sequences considered such as

evolutionary information, allowing to produce much more precise and sophisticated models.

the sequence profiles have been created for both the train set and the blins test set, proceeding in two steps:

* The database containing the UniProtKB/SwissProt sequences has been downloaded. To make this database usable by PSI-BLAST it was necessary to build an index of this sequences using makeblastdb[15].
* finally, PSI-BLAST[16] was performed on each sequence of both the train and the blind set, by inserting specific parameters, thus obtaining the sequence profiles.

The parameters used for the PSI-BLAST run are:

* number of maximum iterations = 3
* e-value threshold = 0.01

after the steps mentioned above, the profiles that did not give reliable results were eliminated, thus obtaining 1204 sequence profiles for the train set (114 removed), and 138 for the blind test set (12 removed).

To compensate for lost profiles in the blind test set, one-hot sequence profiles have been created.

# GOR method

The Garnier-Osguthorpe-Robson method takes its name from its 3 developers, who released it in 1978.

Together with the Chou-Fasman method constitutes one of the pillars of the second generation secondary structure prediction.

Unlike the previous methods, in both cases the prediction of the secondary structure is carried out considering not only the single residue, but also the influence of the neighboring residues.

over the years the method has been optimized to be able to perform secondary structure prediction on sequence profiles[18].

The GOR method is based on a combination between information theory and Bayesian statistics, and evolutionary information if sequence profiles are implemented.

To understand the mathematical approach behind the method, it is useful to start from the simplified version, which considers the influence of a single residue to predict the conformation (eq.1).

The information function to predict this confarmation is:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

Where P(S|R) is the conditional probability of observing

the secondary structure conformation S when the residue is R (one of the 20 residues).

P(S) is the marginal probability of observing the secondary structure conformation S.

If we apply the chain rule to the previous equation we derive the following formulation (eq.2):

|  |  |
| --- | --- |
|  | (2) |

Obtaining the following parameters that we can estimate from a train set:

* P(R,S): the joint probability of observing the residue type R in conformation S
* P(R): the marginal probability of observing a residue type R
* P(S): the marginal probability of observing a residue in conformation S

If we extend the above formula to apply it to a determined window of the sequence, this becomes (eq.3):

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

Where d is (eq.4):

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

Determining the joint probability is computational very expensive since we have a very large number of possible configurations. To overcome this problem, a simplified assumption is used, in which the residues in the windows are considered statistical independents (eq.5). We can rewrite:

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

If we substitute this assumption on the windows based formula (eq.3) of the GOR we obtain that (eq.6):

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

With this new formula the parameters that will have to be estimated from the train set are:

* P(,S): the probability of observing a conformation S for the central residue and a residue of type R at position k in the window
* P(): the probability of observing a residue of type R at position k in the window
* P(S): the probability of observing the conformation S

Considering the new implementation of the method using the sequence profiles, the informative function change in (eq.7):

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

Where is the corresponding residue frequency in the profile window.

In the prediction phase each residue position of a query sequence is analyzed using the model obtained in the training phase. The predicted conformation of a residue R at a certain position is the one having the highest value of the window-based information function (eq.8):

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

## GOR method implementation

To implement the GOR method two python scripts have been created. The first script take in input a directory that contain the sequence profiles and the DSSP files. The window size can be defined from the user as an argument, in this project we use a window size of 17. The output given by this script is a matrix that contains all the useful information for the prediction phase.

The second script includes the prediction phase, takes as input the previously created model and the sequence profiles from which the secondary structures will then be predicted (*both script are available in supplementary informations*)

# Support Vector Machines

Support Vector Machines (SVMs) are supervised machine learn- ing models used for both classification and regression problems, introduced for the first time in 1992 [19].

SVMs are able to solve linear and non-linear problems through the use of kernels, that map the points in in a so called feature space, making linear separation possible.

Regarding binary classification problems, SVMs works by finding the best separating hyperplane between the two classes, called margin, that maximize the distance. To define the margin it is necessary to identify the most extreme points of the two classes, called support vectors, on which the SVM will base its separation (Fig 7).

Chart

Description automatically generated with medium confidence

Figure 7: the hyperplane is the black line that separate the classes.

If we consider a dataset in wich each element have a value and belong into class , we can define the hyperplane (eq.9) for a linearly separable problem as:

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

The distance from an example to the hyperplane is defined by the following formula (eq.10):

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

The examples closest to the hyperplane are the Support Vectors (SV) and the margin is the distance among the SV of the two classes. In accordance with the previous definitions we can define a SV as (eq.11):

|  |  |  |
| --- | --- | --- |
|  |  | (11) |

After rescaling and by we can rewrite both the margin (eq.10) and the distance between an hyperplane and SV (eq.12):

|  |  |  |
| --- | --- | --- |
|  |  | (12) |

the optimization problem on which SVMs are based is to minimize over , such that .

One of the most common strategies to solve this optimizzation problem involves constructing a dual problem (eq.13) where a Lagrange multiplier ⍺i is associated with every inequality constraint in the original problem, respecting the Karush-Kuhn-Tucker (KKT) conditions (eq.14):

|  |  |  |
| --- | --- | --- |
|  |  | (13) |

|  |  |  |
| --- | --- | --- |
|  |  | (14) |

Considering the previous conditions (eq.14), the solution of the dual problem for the super vectors is (eq.15):

|  |  |  |
| --- | --- | --- |
|  |  | (15) |

With this assumptions the classification function for a new point (eq.16) is:

|  |  |  |
| --- | --- | --- |
|  |  | (16) |

For non-linearly separable problems there are two different approaches.

* The soft margin classificartion is applicable in problems defined “soft non-linearly separable”. In this method The hard margin formulation can be modified to incorporate slack variables , that allow misclassification. The stack variable is controlled by an hyper parameter . The soft margin optimization problem (eq.17) and relatives KKT conditions are:

|  |  |  |
| --- | --- | --- |
|  |  | (17) |

* The “kernel trick” (eq.18) is applied in cases of "hard non-linear problems". With this method the points are re-mapped in a feature space, thus making the classification linearly separable. the Kernel is a function that is equivalent to an inner product in some feature space.:

|  |  |  |
| --- | --- | --- |
|  |  | (18) |

The kernel we use is the radial basis function (RBF), described by the following function (eq.19):

|  |  |  |
| --- | --- | --- |
|  |  | (19) |

* 1. **SVM for secondary structure prediction**

In the case of the prediction of secondary structures, 3 or more classes corresponding to the possible conformations of the structure are taken into consideration. In these cases, a multi-class SVM is implemented using an approach called "one-vs-rest" (OvR), which splits a multi-class classification into one binary classification problem per class. In our case SVM has been implemented with scikit-learn library, a Machine-Learning library in Python based on numpy, scipy and matplotlib.

The train phase consists in specifying the various parameters to be used and then using the "fit (x, y)" function in which "x" is the training vectors and "y" is the target class values

* training vectors: is a matrix that represents the sequence profiles of the data set with shape (n\_samples, n\_features), where in our case "n-samples" are the residues present in all the sequences considered, and "n\_features" is equivalent to a vector of 340 elements that contains the data in a windows range of 17 residuals for each specific position of the sequence profile.
* target class values: is a vector of length equal to the sum of the lengths of the sequences in the dataset, in which each position corresponds to the real secondary structure reported for a specific residue (1 = "H", 2 = "E", 3 = "C")

RBF kernel was used in all models, while the other parameters that have been tested are:

* “” is the Kernel coefficient for RBF
* “C” is the Penalty parameter of the error term

The prediction phase consists in the use of the "predict (x)" function on the dataset obtained in the train phase, in which "x" is the training vectors of the dataset on which the prediction is to be performed.

# Evaluation procedure and scoring measure

For SVM method the process is divided into two steps:

* initially 4 models with different parameters were created in order to evaluate in the 5 fold cross-validation phase (training set) which combination of parameters best approached our problem.

**model1:** C=2,

**model2:** C=2,

**model3:** C=4,

**model4:** C=4,

Once the cross-validation results were obtained, the combinations of parameters that presented the highest mean of Matthews correlation coefficient (MCC, eq.21) was selected for the next step (*the scoring indexes will be discussed in detail*).

* The best combination of parameters will be used to execute the train phase on the entire train-set and then proceed with the prediction phase on the blind-set. In order to obtain reliable indices that allow us to compare SVM and GOR method based on their ability to predict the secondary structure.

the evaluation phase consists in comparing the predictions obtained from the models with the real secondary structures and obtaining indices that allow to evaluate the accuracy of the predictions. The process is divided into several steps:

* being a multi-class problem the first phase consists in the creation of a k-class confusion matrix, where k is the number of classes, in our case:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | predicted | | |
|  |  | H | E | C |
| observed | H |  |  |  |
| E |  |  |  |
| C |  |  |  |

*Table 1: 3-classes confusion matrix*

* Both in the cross validation phase and test phase the 3-classes quality score (eq.20) was calculated ():

|  |  |  |
| --- | --- | --- |
|  |  | (20) |

* To allow a better application of the scoring indexes the matrix has been divided into 3 different 2-class matrices, one for each of our classes ("H", "E", "C"):

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | predicted | |
|  |  | H | non-H |
| observed | H |  |  |
| Non-H |  |  |
|  |  | predicted | |
|  |  | E | non-E |
| observed | E |  |  |
| Non-E |  |  |
|  |  | predicted | |
|  |  | C | non-C |
| observed | C |  |  |
| Non-C |  |  |

Table 2: 2-classes confuzion matrices

Considering the matrices in table 2 we can identify the following parameters:

* C: correct positive
* U: under-predictions
* O: over-predictions
* N: correct negative

Both in the cross validation phase and in the test phase, the following scoring indexes were predicted for each matrix:

* Matthew’s correlation coefficient (MCC, eq.21):

|  |  |  |
| --- | --- | --- |
|  |  | (21) |

* true positive rate (SEN, eq.22):

|  |  |  |
| --- | --- | --- |
|  |  | (22) |

* precision (PPV, eq.23):

|  |  |  |
| --- | --- | --- |
|  |  | (23) |

* Accuracy (ACC, eq.24):

|  |  |  |
| --- | --- | --- |
|  |  | (24) |

For each model on which the cross validation was carried out, the average and the standard error (SE, eq.25) were also calculated for each of the scores mentioned before.

|  |  |  |
| --- | --- | --- |
|  |  | (25) |

# Results

The cross-validation results obtained using the 4 parameter combinations are shown in table 4. Looking at the table we can deduce that the combination of the parameters C=2, returns the highest values ​​for both and MCC in all the secondary structures taken into consideration.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | C = 4 | gamma = 2 | | | C = 4 |gamma= 0,5 | | | C = 2 | gamma = 2 | | | C=2 |gamma = 0,5 | | |
| H | E | C | H | E | C | H | E | C | H | E | C |
| Q3 | 0,4660,008 | | | 0,7020,006 | | | 0,4670,008 | | | 0,7080,006 | | |
| MCC | 0,251  0,016 | 0,104  0,022 | 0,153  0,011 | 0,646  0,017 | 0,485  0,013 | 0,482  0,008 | 0,254  0,017 | 0,106  0,023 | 0,156  0,012 | 0,655  0,016 | 0,494  0,014 | 0,493  0,009 |
| SEN | 0,125  0,011 | 0,020  0,008 | 0,985  0,003 | 0,686  0,021 | 0,399  0,022 | 0,873  0,007 | 0,127  0,012 | 0,021  0,008 | 0,985  0,003 | 0,693  0,021 | 0,410  0,023 | 0,877  0,007 |
| PPV | 0,875  0,029 | 0,808  0,075 | 0,443  0,006 | 0,838  0,010 | 0,788  0,017 | 0,620  0,008 | 0,876  0,030 | 0,820  0,074 | 0,074  0,006 | 0,844  0,010 | 0,792  0,019 | 0,625  0,009 |
| ACC | 0,683  0,013 | 0,782  0,013 | 0,467  0,008 | 0,852  0,010 | 0,843  0,009 | 0,719  0,006 | 0,684  0,013 | 0,783  0,013 | 0,468  0,008 | 0,845  0,010 | 0,846  0,009 | 0,725  0,006 |

Table 3: 5-fold cross validation average results and SE for each parameters combination

Comparing the results obtained by the SVM with the best combination of parameters and the results obtained by the GOR method (table 3), we can see that the SVM method returns higher values ​​for and MCC.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| blind set | GOR | | | SVM | | |
| H | E | C | H | E | C |
| Q3 | 0,610 | | | 0,773 | | |
| MCC | 0,538 | 0,446 | 0,387 | 0,732 | 0,661 | 0,584 |
| SEN | 0,743 | 0,747 | 0.450 | 0,750 | 0,641 | 0,871 |
| PPV | 0,687 | 0,482 | 0,734 | 0,895 | 0,830 | 0,680 |
| ACC | 0,781 | 0,758 | 0,713 | 0,878 | 0,885 | 0,783 |

Table 4: performance on blind test of GOR and SVM methods

The better prediction capacity of the SVM method is due to a much more complex and sophisticated computational system, which however disfavors it in the time required in the train phases.

Instead, the lower prediction capacity recorded by the GOR method may be due to the simplification described in equation 5, in which each residue within the window is considered statistically independent, this assumption does not reflect the biological reality.

We can also observe that the true positive rate (SEN, eq.22) for the class H and E of the GOR method is higher compared to that of the SVM, both in the cross-validation phase (table 5) and on the prediction of the blind-test, despite this, the precision of the prediction of these classes (PPV, eq.23) assume very low values, which shows that the GOR method in our case tends to over predict these classes.

Both methods have a greater predictive capacity for helical structures, the difference between helical structures (H) and strand structures (E) can be due to the disparity of these classes in the initial datasets, as can be seen in the figure 6.

|  |  |  |  |
| --- | --- | --- | --- |
|  | GOR cross validation | | |
| H | E | C |
| Q3 | 0,6150,001 | | |
| MCC | 0,5230,002 | 0,4350,002 | 0,4140,001 |
| SEN | 0,8030,002 | 0,7040,004 | 0,4280,002 |
| PPV | 0,6310,003 | 0,4830,004 | 0,8020,002 |
| ACC | 0,7630,001 | 0,7680,000 | 0,7130,001 |

Table 5: 5-fold cross validation average result and SE for GOR model

# Conclusion

Although the two methods generalize very well in never-before-seen datasets, the work carried out has allowed us to clarify with certainty which of the two methods taken into consideration has a greater ability to predict the secondary structure (table 4), taking into account different scoring indexes in order to have a more detailed view of the results. However, both methods have large margins for improvement in the implementation phase.

* Consider minimum length of residues to form both helix and strand conformation.
* Increase the number of hyperparameters taken into consideration and perform a more precise grid search.
* Increase the size of the window.

Acknowledgements

References

1. Rost B. Review: protein secondary structure prediction continues to rise. J Struct Biol. 2001 May-Jun;134(2-3):204-18. doi: 10.1006/jsbi.2001.4336. PMID: 11551180.
2. Chou PY, Fasman GD. Prediction of protein conformation. Biochemistry. 1974 Jan 15;13(2):222-45. doi: 10.1021/bi00699a002. PMID: 4358940.
3. Garnier J, Osguthorpe DJ, Robson B. Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. J Mol Biol. 1978 Mar 25;120(1):97-120. doi: 10.1016/0022-2836(78)90297-8. PMID: 642007.
4. Lim VI. Algorithms for prediction of alpha-helical and beta-structural regions in globular proteins. J Mol Biol. 1974 Oct 5;88(4):873-94. doi: 10.1016/0022-2836(74)90405-7. PMID: 4427384.
5. Muggleton S, King RD, Sternberg MJ. Protein secondary structure prediction using logic-based machine learning. Protein Eng. 1992 Oct;5(7):647-57. doi: 10.1093/protein/5.7.647. Erratum in: Protein Eng 1993 Jul;6(5):549. PMID: 1480619.
6. Zvelebil MJ, Barton GJ, Taylor WR, Sternberg MJ. Prediction of protein secondary structure and active sites using the alignment of homologous sequences. J Mol Biol. 1987 Jun 20;195(4):957-61. doi: 10.1016/0022-2836(87)90501-8. PMID: 3656439.
7. Hua S, Sun Z. A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. J Mol Biol. 2001 Apr 27;308(2):397-407. doi: 10.1006/jmbi.2001.4580. PMID: 11327775.
8. Aydin Z, Altunbasak Y, Borodovsky M. Protein secondary structure prediction for a single-sequence using hidden semi-Markov models. BMC Bioinformatics. 2006 Mar 30;7:178. doi: 10.1186/1471-2105-7-178. PMID: 16571137; PMCID: PMC1479840.
9. D. T. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, 292(2):195–202, September 1999. doi:10.1006/jmbi.1999.3091.
10. Drozdetskiy A, Cole C, Procter J, Barton GJ. JPred4: a protein secondary structure prediction server. *Nucleic Acids Res*. 2015;43(W1):W389-W394. doi:10.1093/nar/gkv332
11. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990 Oct 5;215(3):403-10. doi: 10.1016/S0022-2836(05)80360-2. PMID: 2231712.
12. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, Madden TL. BLAST+: architecture and applications. BMC Bioinformatics. 2009 Dec 15;10:421. doi: 10.1186/1471-2105-10-421. PMID: 20003500; PMCID: PMC2803857.
13. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 1997 Sep 1;25(17):3389-402. doi: 10.1093/nar/25.17.3389. PMID: 9254694; PMCID: PMC146917.
14. Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers. 1983 Dec;22(12):2577-637. doi: 10.1002/bip.360221211. PMID: 6667333.
15. BLAST® Command Line Applications User Manual [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); 2008-. Building a BLAST database with your (local) sequences. 2008 Jun 23 [Updated 2021 Jan 7].Available from: https://www.ncbi.nlm.nih.gov/books/NBK569841/
16. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 1997 Sep 1;25(17):3389-402. doi: 10.1093/nar/25.17.3389. PMID: 9254694; PMCID: PMC146917.
17. Garnier J, Osguthorpe DJ, Robson B. Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. J Mol Biol. 1978 Mar 25;120(1):97-120. doi: 10.1016/0022-2836(78)90297-8. PMID: 642007.
18. Kloczkowski A, Ting KL, Jernigan RL, Garnier J. Combining the GOR V algorithm with evolutionary information for protein secondary structure prediction from amino acid sequence. Proteins. 2002 Nov 1;49(2):154-66. doi: 10.1002/prot.10181. PMID: 12210997.
19. Cortes, C., Vapnik, V. Support-vector networks. *Mach Learn* **20,**273–297 (1995). https://doi.org/10.1007/BF00994018