Chapter 1

Introduction

The protein secondary structure prediction is an essential problem in bioinformatics. The structure mostly depends on the primary amino acid sequence of the protein. Secondary structure prediction belongs to the group of pattern recognition and classification problems. The secondary structure of a given instance is predicted based on its sequence features. One of the known solutions is using the Support Vector Machine (SVM) to predict the secondary structure, which has been described in [2] and [3]. The aim of this work was the implementation of the protein secondary structure predictor based on a logistic regression model. To do this we implemented the algorithm described in the mentioned articles. The project was implemented in R programming language. The source code is available on GitHub: https://github.com/julimer228/ProteinSecondaryStructurePrediction.

Chapter 2

Methods

2.1 Dataset

The dataset consists of three text files: the training dataset, the testing dataset and the validation dataset. Each file has the following structure: the first line provides information about the sequence identification code. The second line contains the sequence of amino acids. In the third line, the secondary protein structure is written. The proteins are separated with an empty line. There are no missing values in this dataset.

2.1.1 Working with the large dataset

Due to the large size of the dataset, we decided to use R libraries that allowed us to perform calculations on multiple cores: parallel and doParallel. In addition, we saved the trained binary classifiers in RDS files. This operation allowed us to remove models from the workspace and free the memory, which was very important for performing further calculations. We used google drive to store the models, which also allowed us to transfer data between two computers.

2.2 Measures

To evaluate achieved results we used two commonly used measures Q3 and SOV.

2.2.1 Q3

The secondary structure prediction is usually evaluated by Q3 accuracy, which measures the percent of residues for which a 3-state secondary structure is correctly predicted.

2.2.2 SOV

The segment overlap score (SOV) [4] is used to evaluate the predicted protein secondary structures, a sequence composed of helix (H), strand (E), and coil (C), by comparing it with the native or reference secondary structures. The main advantage of SOV is that it can consider the size of continuous overlapping segments and assign extra allowance to longer continuous overlapping segments instead of only judging from the percentage of overlapping individual positions as Q3 score does.

2.3 Logistic Regression

Logistic regression analyzes the relationship between multiple independent variables and a categorical dependent variable and estimates the probability of occurrence of an event by fitting data to a logistic curve. Binary logistic regression is commonly used when the outcome variable is binary and the predictor variables are either continuous or categorical. The logistic regression model is based on a logistic function of the form:

$$f(z) = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}} \tag{2.1}$$

The function is presented in the figure 2.1 [1].

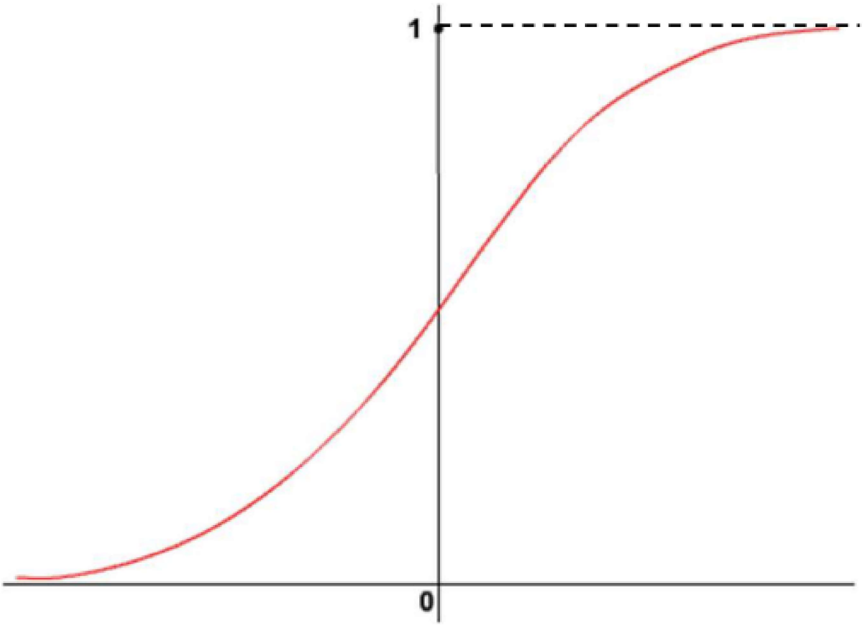


Figure 2.1: The logistic curve.

The logistic function takes values from 0 (when x goes to minus infinity) to 1 (when x goes to plus infinity). It takes values from 0 to 1. The model can describe probability values, determining, for example, the risk of disease or chance of recovery, or, as in our case, membership in a selected class (a detailed explanation of classifiers is given later in the report). The shape of the function resembles an outstretched letter S. This means that until a certain threshold value, the changes in the value of the function are minimal, then rapidly increase to 1. and remain at a very high level (close to 1). In the project, we assumed a threshold value of 0.5.

Let Y denote a dichotomous variable, taking values: 1 if a sample belongs to class C and 0 if the sample does not belong to class C. Then the logistic regression model can be defined with the equation:

$$P(Y = 1|x_1, x_2, \ldots, x_k) = P(X) = \frac{e^{a_0 + \sum_{i=1}^{k} a_i x_i}}{1 + e^{a_0 + \sum_{i=1}^{k} a_i x_i}} \tag{2.2}$$

Where:
$a_i, i = 0, \ldots, k$ - are regression coefficients,
$x_1, x_2, \ldots, x_k$ - are independent variables.

The left side of the equation is the conditional probability that the variable Y will take a value equal to 1 for independent values $x_1, x_2, \ldots, x_k$.

2.4 Algorithm Description

2.4.1 Sliding window coding scheme

After investigating the dataset, we created the input groups for the logistic regression classifier. We followed the instructions described in [2]. First, we had to implement the sliding window scheme. This method allows to preserve the information about the local interactions among neighbouring residues. In the beginning, we had to choose the size of the window. To predict the structure of the amino acid in the middle we need to use the sequence whose size is equal to the size of the window. As can be seen, the problem of missing amino acids at the ends of the sequence had to be solved. We complemented the missing values with the empty character "-" to keep the correct window size. The described algorithm for the window size 5 is presented in the image 2.2.

2.4.2 Orthogonal Input profile

The next step was to use orthogonal encoding to assign a unique binary vector to each residue. The weights of all the residues in the window have the value 1 and the rest have the value 0. As a result, we obtain the input matrix with 21 columns (there are 20 different amino acids and one value assigned to the empty character) and with a number of rows equal to the size of the window. Next, we reshaped the orthogonal input to get the one-dimensional input vector. The size of the vector is equal to the result of the multiplication of the number of rows of the matrix by the number of columns. The

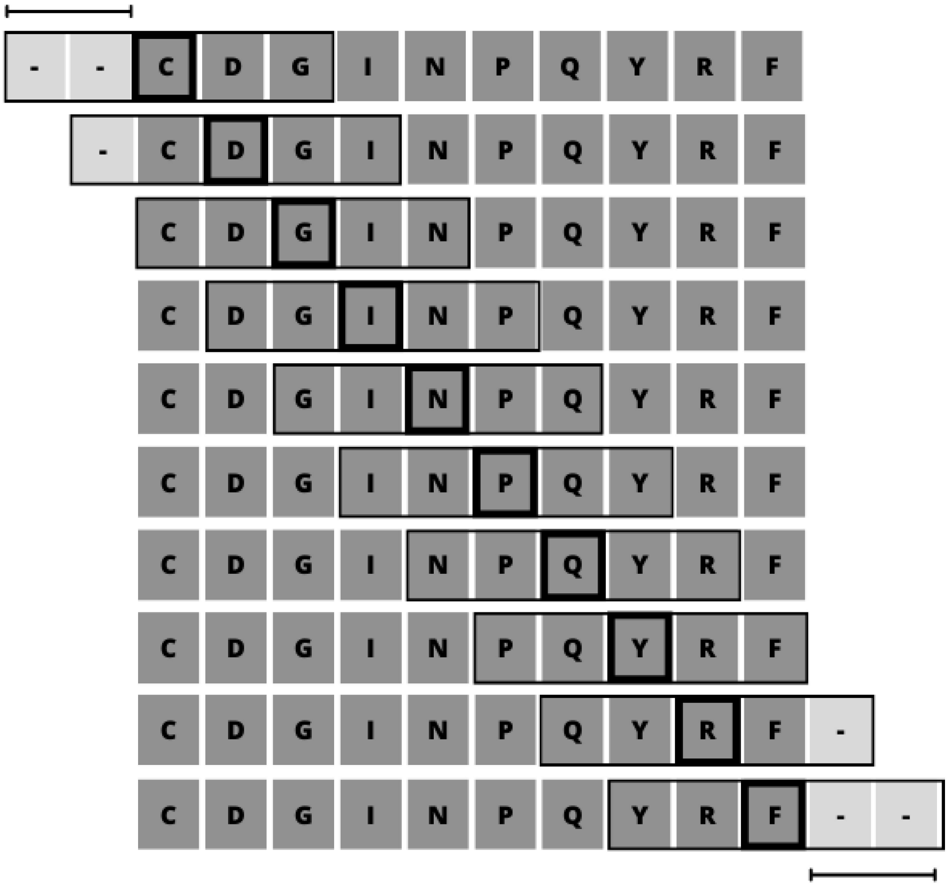


Figure 2.2: The sliding window scheme.

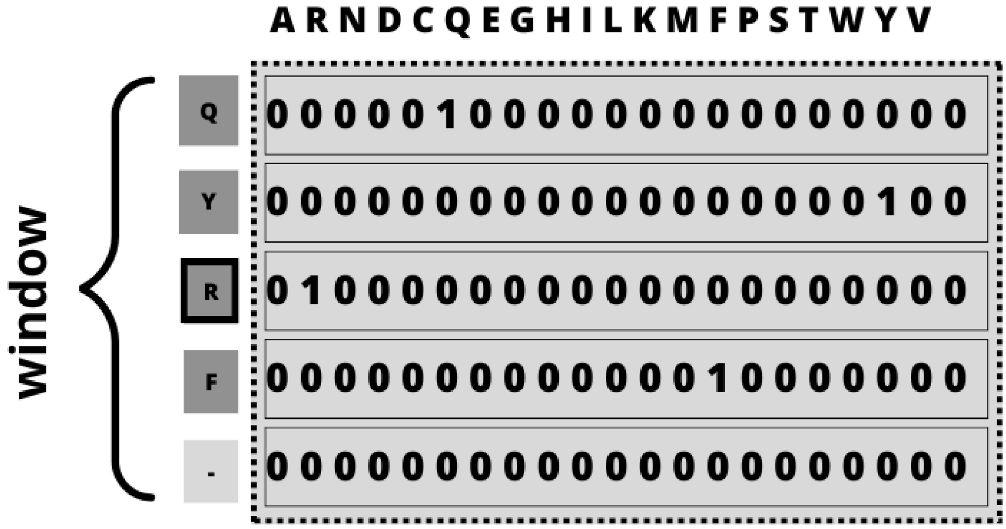
following rows were written to the vector one by one. Encoding for the input window of the size 5 is presented in the image 2.3

2.4.3 Constructing the binary classifiers

We constructed six binary classifiers: three one-versus-one classifiers (H/E, C/E, C/H) and three one-versus-rest classifiers (C/∼C, E/∼E, H/∼H). For each classifier, we trained the logistic regression model.

2.4.4 Constructing tertiary classifier

The binary classifiers were used to create the different tertiary classifiers. We created three tree classifiers described in [3] (C/∼C & H/E, E/∼E & C/H, H/∼H & C/E). For example for the second classifier when the first binary classifier classifies the sample as C its predicted value is C, otherwise the class is predicted by the second one-versus-one classifier H/E. Their structures are presented in figures 2.4, 2.5, 2.6. We also tested the classifier based on three one-versus-one classifiers (C/∼C & E/∼E & H/∼H). The sample is as-



	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
Q	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2.3: Orthogonal encoding.

signed the class with the highest probability.

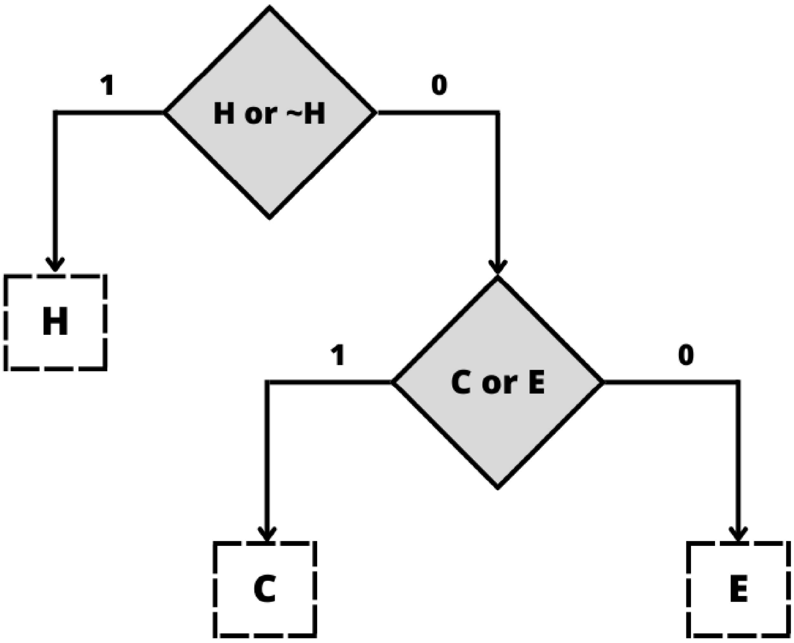


Figure 2.4: H/∼H & C/E

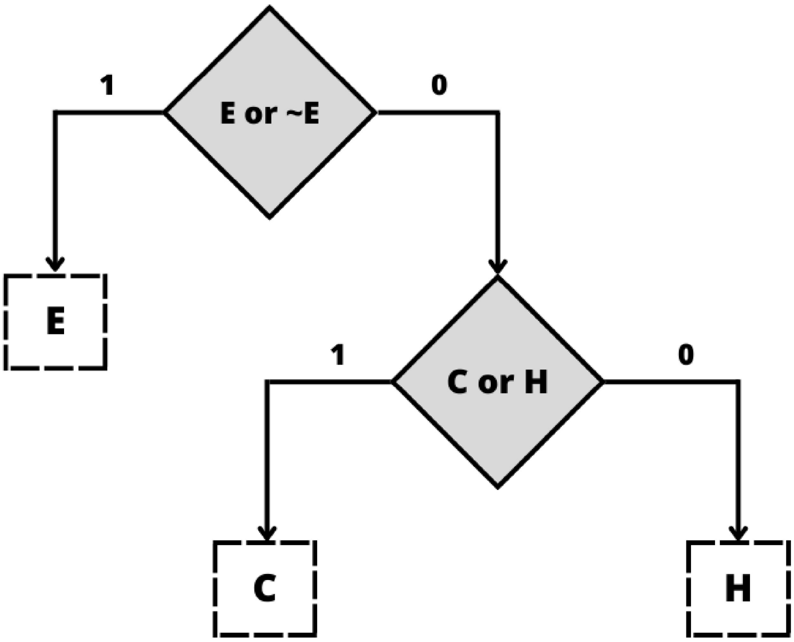


Figure 2.5: E/∼E & C/H

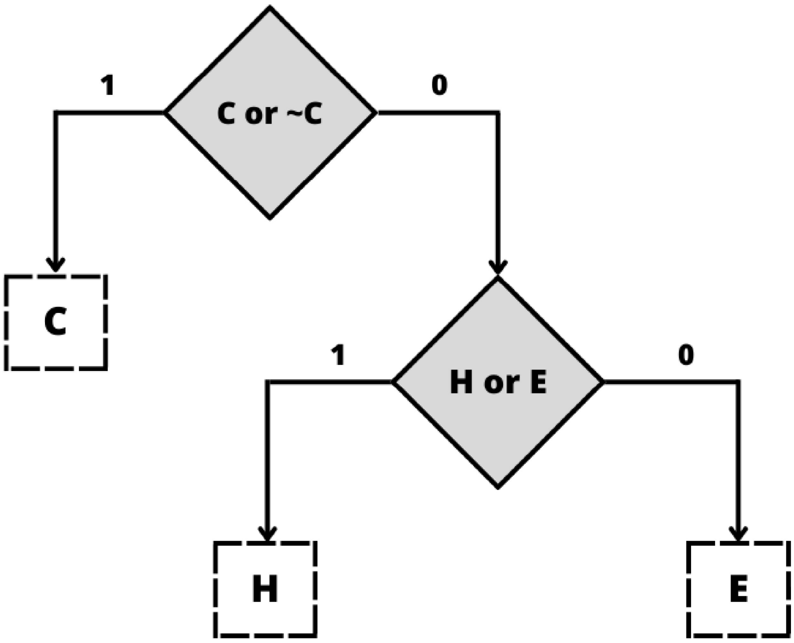


Figure 2.6: C/∼C & H/E

Chapter 3

Results

We tested different window sizes (from 5 to 13 amino acids, only odd sizes). The table 3.1 presents the accuracy scores obtained for each binary classifier on the test dataset. The best results were obtained for the window of size 13.

Window Size	5	7	9	11	13
C/∼C	72.48	72.97	73.06	73.17	**73.27**
E/∼E	78.36	78.68	79.04	79.35	**79.61**
H/∼H	69.95	71.55	72.33	72.75	**73.11**
H/E	69.75	71.37	72.47	73.41	**74.09**
C/E	74.40	74.87	75.02	75.30	**75.53**
C/H	72.34	71.37	73.85	74.03	**74.30**

Table 3.1: Accuracy for each binary classifier. Data was sampled with different window sizes.

Then we compared the Q3 and SOV results we got for each of the tertiary classifiers. To do that, we saved the predicted structure in a FASTA format. For each classifier, we used the window that provided the best accuracy for the binary classification. Results are presented in the table 3.2. The best results were achieved by the H/∼H & C/∼C & E/∼E classifier.

Classifier	Q3	SOV
H/∼H & C/∼C & E/∼E	**61.79**	**56.10**
H/∼H & C/E	60.66	54.40
E/∼E & C/H	60.98	54.80
C/∼C & H/E	60.66	55.20

Table 3.2: Q3 accuracy and SOV for each tertiary classifier.

Chapter 4

Conclusions

The project allowed us to learn about the problem of protein secondary structure prediction. The obtained results are worse than the results achieved with the method described in the [3]. It should be noted that the dataset used in the project is not exactly the same dataset used by the authors of the articles. However, the Q3 and SOV measures are better than 33.33%, so our classifiers work better than guessing. Testing different window sizes allowed us to find the binary classifiers that provide better accuracy. Building the classifier from three one-versus-rest classifiers allowed us to achieve the highest Q3 and SOV values. In order to further develop the project, it would be necessary to see if larger window sizes would allow for greater model accuracy, as it can be seen that for the tested window sizes, accuracy increases as the size increases. One could also try to build more complex classifiers for recognizing the three classes, possibly to improve prediction accuracy. The Current solutions based on deep learning provide better SOV and Q3 measures than our solution based on logistic regression.

Bibliography

[1] URL: https://home.agh.edu.pl/~mmd/_media/dydaktyka/adp/regresja_logistyczna.pdf.

[2] Mayuri Patel and Hitesh Shah. 'Protein Secondary Structure Prediction Using Support Vector Machines (SVMs)'. In: *2013 International Conference on Machine Intelligence and Research Advancement*. 2013, pp. 594–598. DOI: 10.1109/ICMIRA.2013.124.

[3] Hua Sujun, and Sun Zhirong. 'A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach'. In: *Journal of molecular biology*. 2001, 397—407.

[4] Liu T and Wang Z. 'A further refined definition of segment overlap score and its significance for protein structure similarity'. In: vol. Source Code Biol Med. 2018, pp. 594–598. DOI: 10.1109/ICMIRA.2013.124.

Appendices

List of Figures

List of Tables