Evaluation of machine learning models for classification of enzymes to functional classes

Ana Jerina Faculty of Computer
and Information Science
University of Ljubljana
Večna pot 113, 1000 Ljubljana
Email: aj1801@student.uni-lj.si

Abstract—Enzymes are proteins that carry an important role in metabolic processes in organisms by catalyzing biochemical reactions. There are 7 main functional classes, but their precise classification is based on their exact function, which is denoted by EC number. As for all proteins, behaviour of enzymes is heavily influenced by their amino acid sequence. There are 20 different amino acids that represents the basic building blocks of enzymes.
Empirical identification of enzyme function is costly, slow and complex, making it a problem for which machine learning could be well-suited to assist with. Here support vector machine, k-nearest neighbours, random forest and convolutional neural network are evaluated as enzyme function classifiers.

Index Terms—SVM, kNN, random forest, neural network

1 Introduction

Enzyme function can be determined by assays in laboratories by seeking to what kind of substrate the enzyme reacts to and what kind of products are formed during the process. Assays can require a lot of time and can be expensive. Beside an assay kit, expensive laboratory equipment (such as spectrophotometer) is essential for proper testing.

Machine learning has made a great impact on industry and science by contributing new methods able to find patterns in data and then further use those patterns to perform tasks such as classification or regression. Here classification of enzymes to their main functional classes using a couple different models is presented and evaluated.

December 27, 2022

1.1 What are enzymes

1.1.1 Structure

Predominantly enzymes are proteins, thus their primary structure is a poly-peptide chain (that is a sequence of amino acids linked by peptide bonds). Some of them require either a certain metallic compound or non-protein organic molecule (coenzyme) to activate their use.

Ribozymes (or RNA enzymes) are a small group of enzymes made of ribonucleic acids instead of amino acids. They are located in ribosomes and catalyze linking of amino acids during protein creation. Ribozymes are excluded from this work, since it is based on poly-peptide chain features.

1.1.2 Mechanism

Every enzyme possesses an active site, that is a part of enzyme where the corresponding substrate binds to during the catalyzed process. Substrate needs to be of precise size and shape and must have specific compounds in order to be a right fit for the enzyme (lock and key principle).

As substrate binds to enzyme, the enzyme changes its shape and biochemical reaction takes place. Product is unbound from enzyme upon completion of reaction. Enzyme is then

reused. Generalized functions of main classes are listed in Table 1.

1.1.3 EC number

Enzyme's function is defined by its EC (enzyme commission) number. Each EC number consists of four integers separated by dots. First integer defines to which main class enzyme belongs to (as shown in Table 1), second and third define sub-classes and the last one describes the substrate that binds to enzyme.

There are over 5000 different EC numbers and for a full four-layer classification a tree-structured model would be appropriate. Here for simplification only the main class is handled.

EC	Class	Function
1	oxidoreductases	oxidoreduction
2	transferases	transfering functional groups
3	hydrolases	hydrolysis
4	lyases	adding or undoung bonds
5	isomerases	isomeration
6	ligases	binding substrates
7	translocases	assisting in molecule transfering

Table 1. : Function of enzyme classes

1.2 Data extraction

SwissProt (part of UniProtKB [5]) is a protein database that contains annotated protein data, which was used as a source database for this project.

Acquired data contained 568 744 samples and was reduced as described in EzyPred [1] - only those proteins that met the following requirements were parsed from SwissProt and were used as the basis for data extraction:

- protein has EC number (meaning it belongs to enzymes)
- enzyme contains a single EC number (limit to enzymes with only one function),
- sequence includes between 50 and 5000 amino acids (in order to be a representative sequence),
- sample is not annotated as sequence fragment,

After extraction of such enzymes (256 484 samples) sequence clustering program [3] was used to reduce the data set. Sequences of the same functional class were clustered together in groups of sequences that are over 40% identical. From each cluster only the referential sequence was taken. At this point data set contained 24 614 samples.

Since different models require unequal data formats from here on data preparation for supervised learning and deep learning was done separately.

1.3 Data preparation for supervised learning

Amino acid alphabet contains 20 letters, however at certain positions sequences can contain some other letter that represents presence of undefined amino acid from a subgroup of all amino acids. Protlearn library is unable to handle sequences with undefined amino acids. Since there was only 76 of such enzymes in the extracted data, they were removed from the data set before extracting features.

Features were extracted using protlearn library. Each sequence was presented as its amino acid composition, conjoined triad description and dipeptide composition.

1.3.1 Amino acid composition

computed frequencies for each of 20 amino acids, provides 20 features for each sample.

1.3.2 Conjoined triad description

amino acids are grouped into classes determined by their side chain volume and dipoles. Frequencies of all class-combinations of length 3 sub-sequences are then computed. Since there are 7 different classes each CTD represents each sample with 343 features.

1.3.3 Dipeptide composition

computed frequencies of all possible pairs of amino acids, presenting each sample with 400 features.

All features were then concatenated and each one of 24 538 samples was described by 763 features. This data set was used for supervised learning models.

Oversampling is a process in which less-represented classes get re-sampled in order to even out the class sizes. This data set was also over-sampled to inspect the difference in performance, since class representation is uneven (as presented in Figure 2) Over-sampling could however lead to over-fitting, so caution is necessary.

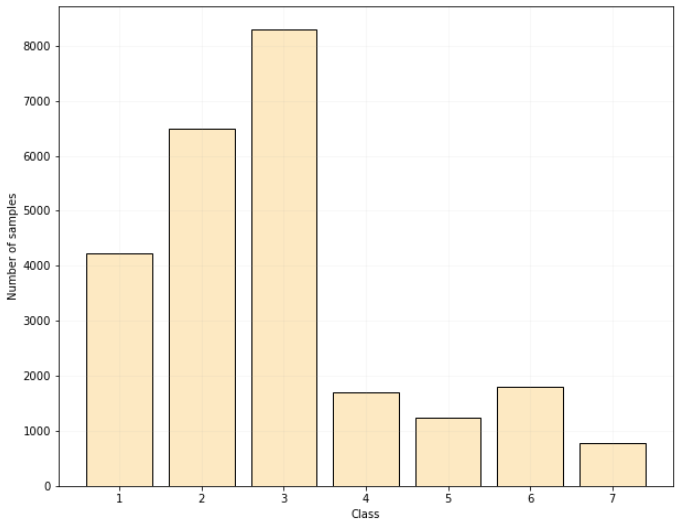

Figure 2. : Number of samples in data set by class

1.4 Data preparation for deep learning

For neural network model each sequence was presented as two matrices, both of size *sequence-length* x 20:

1.4.1 One-hot encoded sequence

Each column of this matrix represented a specific amino acid and rows represented an amino acid at specific location in the sequence. Each row contained zeros everywhere except at the position where amino acid from sequence matches amino acid from column - in such locations the matrix contained 1.

1.4.2 Position-specific scoring matrix

PSSM was acquired by running PSI-BLAST [4] with 3 iterations and e-value 0.002. It represents evolutionary sequence features. More specifically during PSI-BLAST sequences that are most similar to the referential (sample) sequence are found and used to construct the matrix. Each row represents one of amino acids in the referential sequence while columns portrait the 20 amino acids. Sequences are aligned altogether and for each position frequencies of amino acids from all sequences are computed and inserted into corresponding position in PSSM.

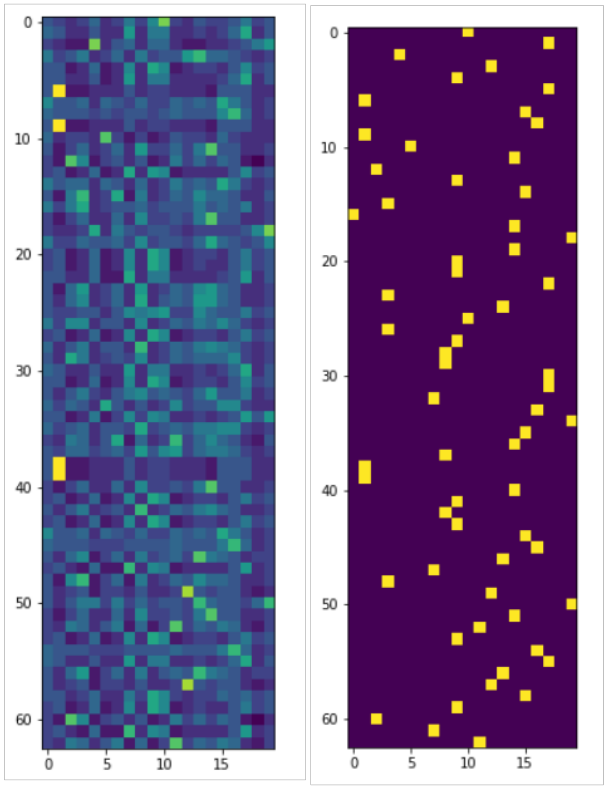

Figure 3. : Example of visualized PSSM (left) and one-hot encoded sequence (right)

2 Brief model description

Models were implemented with Scikit-learn and Pytorch libraries in python.

2.1 Majority classifier

This classifier checks which class is most common in a train set and considers it the only important class. Hence, regardless of the features every new sample that is being classified is given to the most common class. To some extent, it defines a border for other classifiers if they have found any new and better patterns.

The model was done only with original (not over-sampled) data and was tested with 10-fold cross validation.

2.2 SVM

SVM classifier works by finding a map that transforms training data in such a way that the gaps between different classes are as large as possible. The map is then used to transform new samples and the model decides what classes they belong to depending on where in space they end up.

Model was tested on original and over-sampled data and was tested with 10-fold cross validation.

2.3 kNN

K-nearest neighbours classifier checks which rows (k of them) from train data set are closest to the sample by Euclidean distance. The predicted class for a new sample is then decided as a most common class in closest k-rows.

For model construction 5 was used as k on both original and over-sampled data. It was tested with 10-fold cross validation.

2.4 Random forest

Random forest is based on a number of decision trees that get trained using bagging. Bagging helps with stabilizing the model and lowers the chances of over-fitting. When decision trees are getting built each gets trained on its own data set that is a random sample of given data set sampled with replacement and has the same size as the given data set.

When classifying new samples, each tree predicts on its own and the class that got predicted by most trees is presented as final prediction.

Here, the forest contained 100 trees on both original and over-sampled data and was tested with 10-fold cross validation.

2.5 Neural network

Neural networks are built by interlaced neurons organized in layers that take many inputs and return a single output.

In convolutional layer a two- or three-dimensional matrix is processed to a feature map. Pooling layers may be used in between convolutional layers to reduce the size of the data. LSTM is a type of recurrent neural network that is able to preserve long-term interrelationships and can be used as feature extraction tool.

As mentioned above this neural network model works with data set that represents each sample with 2 matrices of size *sequence-length* x 20. Each matrix is processed by 2 convolutional layers each followed by a pooling layer, a LSTM (long-short term memory) layer and finally a fully connected layer to extract features.

Features from both matrices were then concatenated and another fully connected layer was applied. Cross entropy was used as loss function and Adam as optimization function.

3 Results

Figures 5, 6, 8, 10 show visualized confusion matrices of model performances on original data sets and Figures 7, 9, 12 show visualized confusion matrices on over-sampled data.

In Table 4 accuracy and standard error for each model is presented. As mentioned above, all models except for neural network are tested with 10-fold cross validation. Neural network was tested with 5-fold cross validation in order to shorten the runtime.

Model	Accuracy [%]	SE [%]
Majority classifier	33.9	3.674
SVM	82.359	0.971
SVM (oversampled)	98.821	0.464
kNN	40.739	1.165
kNN (oversampled)	71.621	2.145
RF	76.075	0.845
RF (oversampled)	97.435	1.055
Neural network	31.05	6.92

Table 4. : Accuracy and error of models tested with cross validation

3.1 Majority classifier

As Table 2 hints the most common class in data set is hydrolases (class 3). Since all other classes were disregarded by the classifier we only get as many hits as there is actual hydrolases in the test set.

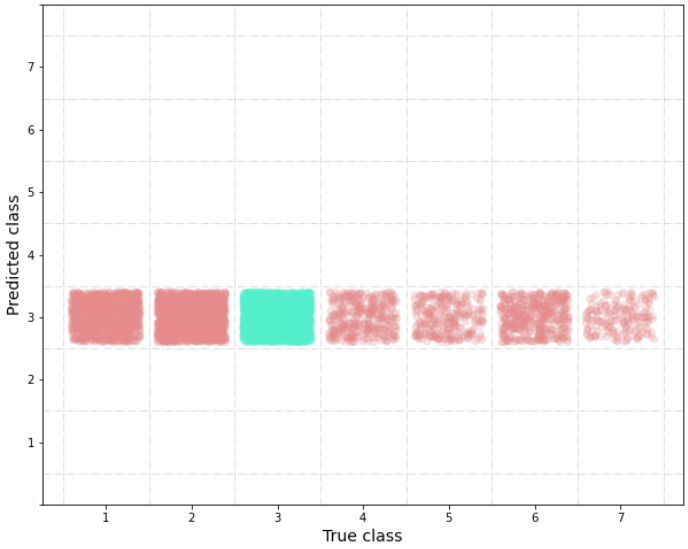

Figure 5. : Performance of majority classifier

3.2 SVM

SVM achieved highest accuracy on both original and over-sampled data set. Most problems arrived when classifying lyases (class 4). Majority of them got classified as hydrolases. On over-sampled data performance improved significantly.

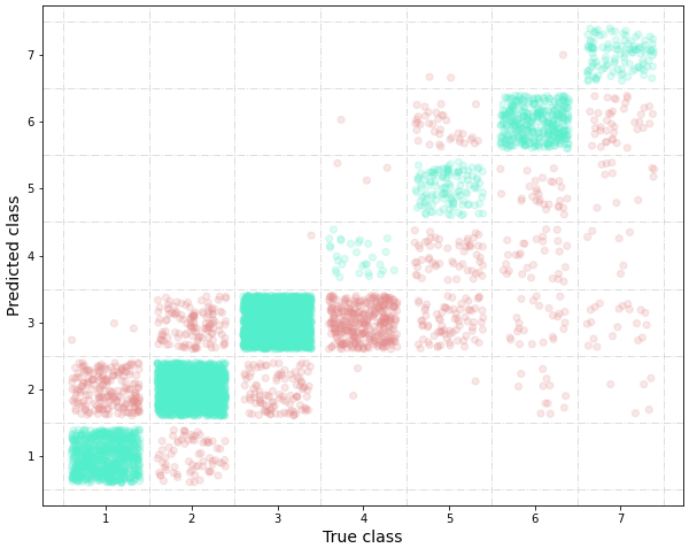

Figure 6. : Performance of SVM classifier on original data

3.3 kNN

KNN model noticeably leaned towards classification into more represented classes.

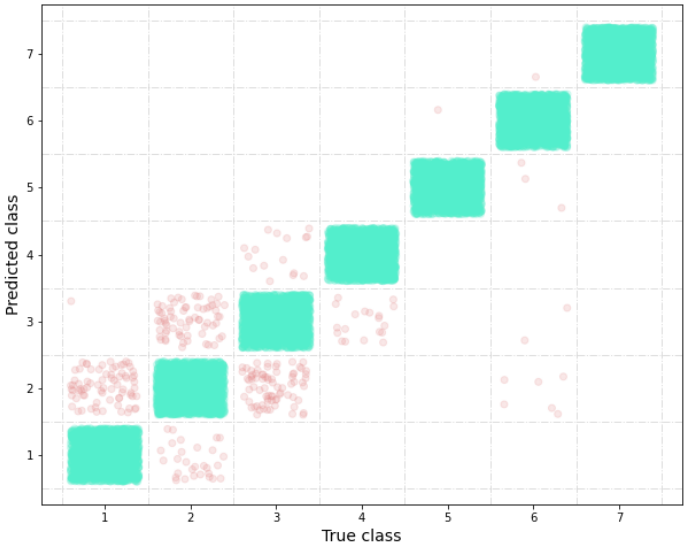

Figure 7. : Performance of SVM classifier on over-sampled data

Over-sampling improved classification overall, but mostly for smaller classes.

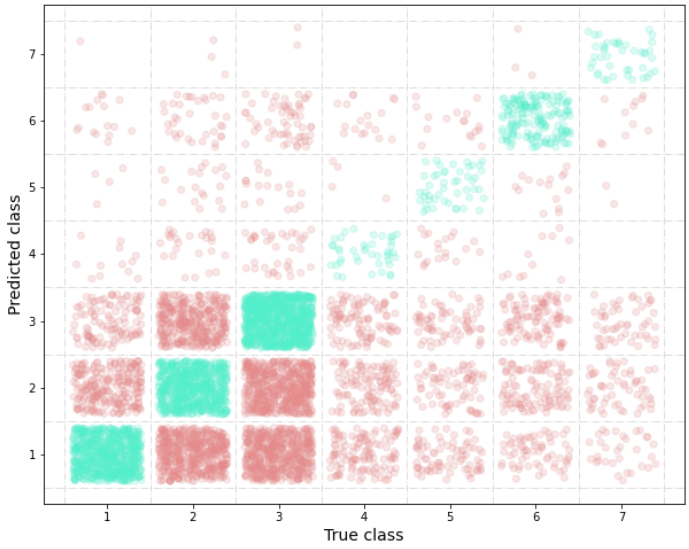

Figure 8. : Performance of k-nearest neighbours on original data

3.4 RF

It is evident that the large classes got classified impressively while most of the less-represented classes often got classified to the largest class.

Over-sampling again improves classification, most often mistake was confusing classes 2 and 3 more.

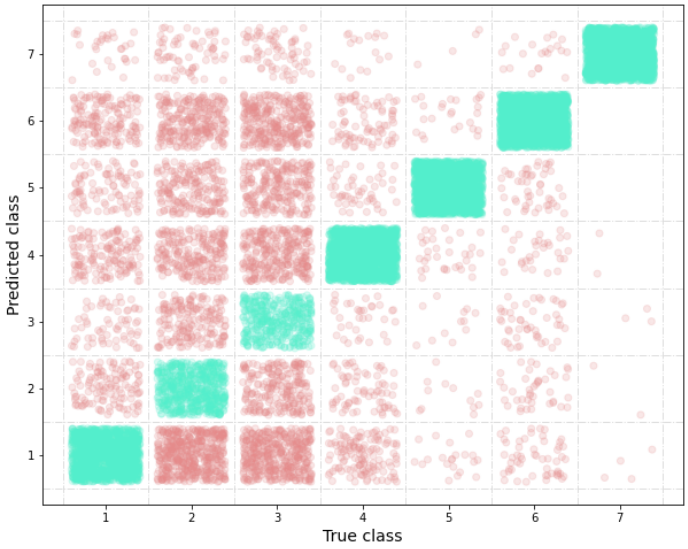

Figure 9. : Performance of k-nearest neighbours on over-sampled data

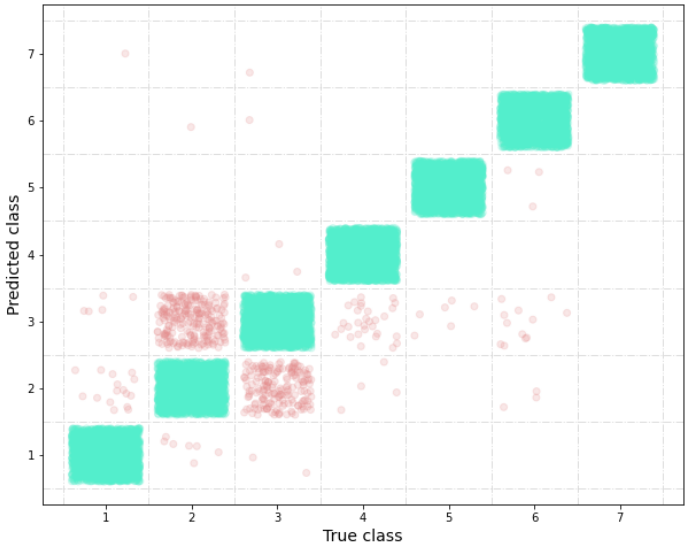

Figure 11. : Performance of random forest on over-sampled data

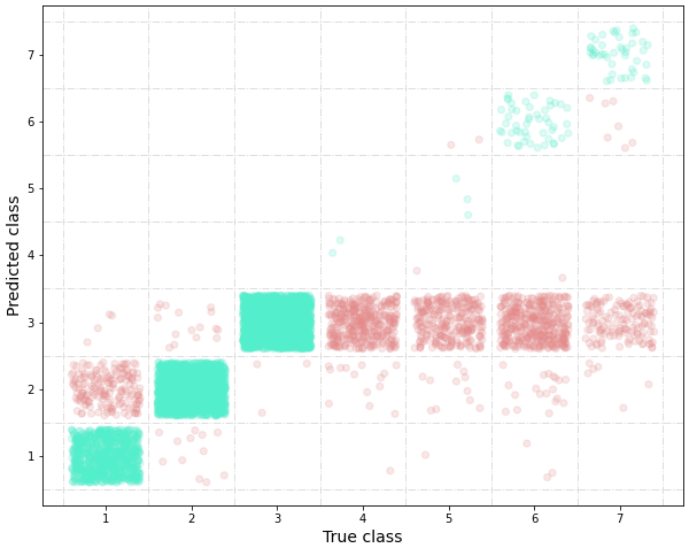

Figure 10. : Performance of random forest on original data

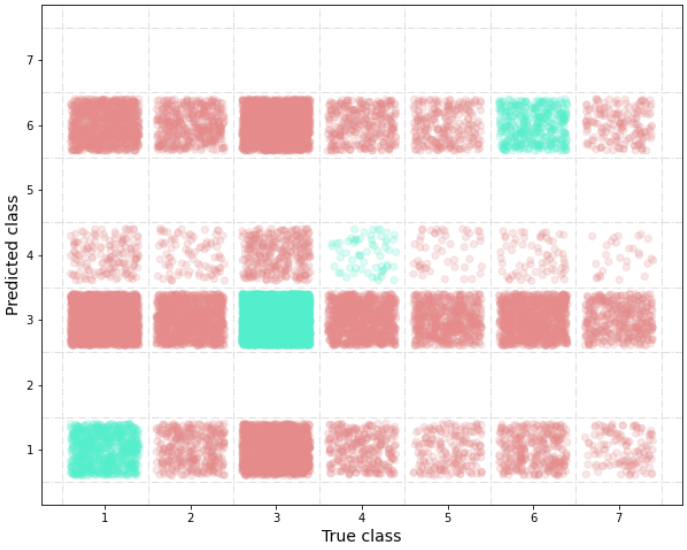

Figure 12. : Performance of neural network

3.5 Neural network

Neural network performed poorly, giving less accurate classifications than majority classifier.

A few classes were never classified in, including transferases which are one of the well-represented classes. The classes that were noticed by classifier got samples from every possible class and there is no noticeable patterns.

4 Conclusion

This was an interesting and useful project to gain some tangible machine learning knowledge and skills. For a first-time attempt to machine learning the outcome is (personally) decent, however there are many things that could be improved in this work - especially with neural network implementation.

First thing that comes to mind about enhancing neural network performance is PSSM matrix computation. In this work BLAST algorithm was run against SwissProt database

instead of against much larger non-redundant protein database. That was due to lack of free space, since running BLAST locally requires the possession of database locally.

The fact that a large class was left out of ever being predicted seems as there might have been a problem with optimization. The loss function value does not decrease, but rather decreases at first and then increases.

More layers could maybe be helpful. Validation set was also missing. Besides that it would be better if neural network was tested with 10-fold cross validation (and not 5-fold) so the comparison between the results would be more reliable.

Overall building the first neural network (even though it was a bad one) was a great learning opportunity.

Regarding testing of all algorithms a second test set could have been created by taking a random sample from all sequences that were disregarded during the clustering part of data reduction.

In future this work could be updated with a tree-structured model that classifies enzymes down to the last layer of EC number.

In every aspect doing this project was definitely beneficial and a time well-spent.

REFERENCES

[1] K.C. Shen and H.C. Chou, *EzyPred: A top–down approach for predicting enzyme functional classes and subclasses*, Biochemical and Biophysical Research Communications, Volume 364, Issue 1, 2007.

[2] Y. Li, S. Wang, R. Umarov, B. Xie, M. Fan, L. Li, X. Gao, *DEEPre: sequence-based enzyme EC number prediction by deep learning.*, Bioinformatics, Volume 34, Issue 5, 2018.

[3] L. Fu, B. Niu, Z. Zhu, S. Wu, W. Li, *CD-HIT: accelerated for clustering the next-generation sequencing data.*, Bioinformatics, Volume 28 Issue 23, 2012.

[4] S.F. Altschul, T.L. Madden, A.A. Schäffer, J. Zhang, Z. Zhang, W. Miller D.J. Lipman *Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.*, Nucleic Acids Research, Volume 25 Issue 17, 2007.

[5] The UniProt Consortium *UniProt: the universal protein knowledgebase in 2021*, Nucleic Acids Research, Volume 49 Issue D1, 2021.

[6] The UniProt Consortium *Determination of enzyme/substrate specificity constants using a multiple substrate ESI-MS assay*, J Am Soc Mass Spectrom, Volume 15 Issue 1, 2005.

[7] Y.H. Li, J.Y. Xu, L. Tao, X. Zeng, S. Li, X.F. Li, P. Zhang, *DSVM-Prot 2016: A Web-Server for Machine Learning Prediction of Protein Functional Families from Sequence Irrespective of Similarity*, PLOS ONE, Volume 11, Issue 8, 2016.