PREDICTION OF TYROSINASE INHIBITORY PEPTIDES USING MACHINE LEARNING APPROACH

Dilrabo Kodirova

Academic Advisor: Professor Jung Hu

INTRODUCTION

Tyrosinase is an enzyme vital for the production of melanin and is linked to an individual's susceptibility to hyperpigmentation disorders, which involves the development of darker, discolored patches of skin. Tyrosinase inhibitory peptides (TIPs) are small peptides with a length of 3-20 amino acid residues, demonstrating tyrosinase inhibitory activity. These findings have significance for pharmaceutical and clinical research, since they can be used as drugs based on the TIPs in the treatment of disorders of hyperpigmentation. It is, however, expensive and time-consuming to identify the possible TIPs through experiments. Therefore, this calls for a need for computational methods that will properly and at lower costs identify TIPs. In this study, our core approach will be to look into the application of machine learning algorithms to train a computational model, able to classify between TIPs and non-TIPs using only peptide sequence information. Thus, more effective and affordable tools are offered for the description of potential inhibitors against tyrosinase.

DATA

We have obtained data from Proteins & Peptides Mining Lab, which is open access for everyone on the website (https://pmlabstack.pythonanywhere.com/dataset_TIPred). Proteins & Peptides

Mining Lab provides 4 datasets that include information about protein sequences only with the following size distribution:

	Training Data				Independent Data			
#	Dataset #1		Dataset #2		Dataset #3		Dataset #4	
TIP-inhibitory	Positive		Negative		Positive		Negative	
Size	106		106		27		181	
Length	short	long	short	long	short	long	short	long
Size (after division)	36	70	2	104	7	20	6	175

However, the sequence information alone is not enough as a data to predict if it is TIP-inhibitory or not. Therefore, extraction of new features and adding them to the initial dataset was needed. For this we used PseAAC - General Software (http://www.csbio.sjtu.edu.cn/bioinf/PseAAC/) created by Shanghai Jiao Tong University, where given the sequence information and input parameters, one can obtain pseudo amino acid composition information.

However, the amount of features and importance of features we can extract also depends on sequence size, which ranges from 2 to 20 for these datasets. Based on the distribution curves, it was decided to divide all datasets into short and long, where short include sequences of length 2 and long include length of 3 and more. This division resulted in a total of 8 sequence datasets for each of which multiple feature sets will be generated.

FEATURES

PseAAC Software allowed us to generate 3 features sets using the following input parameters:

	Feature Set #1		Feature Set #2		Feature Set #3	
length	short	long	short	long	short	long
Input parameters						
PseAA mode	Type 1		Type 2		Dipeptide-composition	
Amino acid character (*i*)	• Hydrophobicity • Hydrophilicity • Mass • pK1 (alpha-COOH) • pK2 (NH3) • pI (at 25oC)		• Hydrophobicity • Hydrophilicity • Mass • pK1 (alpha-COOH) • pK2 (NH3) • pI (at 25oC)		N/A	

Weight factor	0.5		0.5		N/A	
Lambda parameter (λ)	1	2	1	2	N/A	N/A
Output Features						
Feature numbers formula	$20 + \lambda$		$20 + \lambda i$		21*20	
Feature numbers	21	22	26	32	420	420

MODEL SELECTION

For this problem classification algorithm was necessary because we are deciding between positive or negative outcomes of TIPs. One of the classifiers is Support Vector Machine (SVM) Algorithm, which was chosen as a core model for prediction. It uses a technique called the kernel trick to transform data and then based on these transformations it finds an optimal boundary between the possible outputs. This model works best for finding complex relationships, which is the case of the TIP-inhibitory problem.

We used an algorithm by LIBSVM (https://www.csie.ntu.edu.tw/~cjlin/libsvm/) python package that offers an already implemented SVM algorithm.

SOFTWARE INSTRUCTIONS

To run the program for prediction simply call “python main.py” on the terminal. It will read all of the datasets in the “data” folder, the preparation of which was discussed earlier. Then, it applies the SVM model and returns a prediction report called “report.csv” and actual prediction values stored in the “predictions” folder.

RESULTS AND DISCUSSION

Performance Metrics	Feature Set #1		Feature Set #2		Feature Set #3		Feature Sets Combined	
	short	long	short	long	short	long	short	long
ACC	61.54	91.28	61.54	90.77	53.85	91.28	53.85	90.26
MSE	0.38	0.09	0.38	0.09	0.46	0.09	0.46	0.10
SCC	0.10	0.14	0.10	0.09	N/A	0.14	N/A	0.05

Cross Validation Accuracy	94.74	59.20	94.74	59.20	94.74	60.92	94.74	60.34

All performance evaluation metrics point that for all feature sets, performance of long sequence datasets was better with accuracy within 90-91% across all feature sets. The main factor that might have affected the lower performance of short sequence datasets is the dataset size. Shorter sequence datasets still included much less number of sequences than the longer ones, resulting in fewer data points. This makes it difficult for the model to find the relationships within the dataset and classify correctly. Therefore, accuracy and error is much lower for short sequence datasets.

FUTURE IMPROVEMENTS

Future advancements to consider to further improve the algorithm performance start from obtaining bigger datasets, especially for short sequence datasets. Another thing to consider is feature selection. Even though all of the feature sets yielded similar performance, picking the best predicting/correlating features from each set would improve the performance. These are the changes related to data and features, however, some improvements for the algorithm could be delivered too. For example, using a multi-layer approach for the algorithms is something to consider, as the previous related researches practiced this approach successfully.