Predicting the subcellular location of eukaryotic protein

Sorathorn Chaweewongpongdet

Abstract

Motivation: Due to high development of technologies, big data and machine learning are broadly spread in many research area including bioinformatic. To help researcher understand function of eukaryotic protein, knowing subcellular location of these protein might provide hints to researcher to learn the function. This paper applies XGBoost algorithm, which is successful in structure data, via three main feature groups: categories: amino acid composition, physicochemical properties and Nucleus localization signal.

Methods: Random Search, XGBoost, L2-regularization, Precision, Recall and F1 Score

Result: Using XGBoost with these features, the accuracy is around 89%. In each subcellular location, the accuracy of Cytosolic, Mitochondiral, Nuclear and Secreted proteins are 84%, 77%, 98% and 89%, respectively.

1. Introduction

During the fast-growing development of big data and technologies, bioinformatic area also is a remarkable area which has gained the benefits of these development. As a result of it, there are increasing demand of using machine learning techniques to help scientists understand the biological area. Understanding function of protein is a common question in this field. To answer the question, there are several components that could be considered such as protein subcellular location, components of a protein and properties of a protein. Protein subcellular location is information that provides the location of a protein in a. cell. So, protein subcellular location prediction task might help scientists learn some clue about possible function of a protein.

In this paper, the focused protein subcellular location is four major subcellular locations in eukaryotes: Cytosolic, Mitochondrial, Nuclear and Secreted. Cytosolic is a protein that use in a cell but does not live in any organelles. Mitochondrial is a protein exported to the cell's mitochondria. Nuclear category is a protein which is used in cell's nucleus. Secreted is a protein which is transported out of a cell. There are several techniques which are applied on this problem such as logistic regression, support vector machine and XGBoost by using Python programme to implement the algorithm. This paper extracts features from three major area : amino acids composition, physicochemical properties of amino acid and Nucleus localization signal (NLS).

2. Methods

2.1. Dataset

This paper uses dataset from COMPGI10 Coursework in Fasta format which contains 3,004 records of Cytosolic Proteins, 1,605 records of Secreted Proteins, 1,299 records of Mitochondiral Proteins, 3,314 records of Nuclear Proteins and 20 records of "Blind Test" Proteins of eukaryotes, as shown in Table 1. It is necessary to split the data in to 3 dataset. Firstly, the data is split into training and validation set and test set with 80% and 20%, respectively. Then, the

Table 1 : The number of protein sequence in each subcellular

Organism	Subcellular Location	Number of proteins
Eukaryotes	Cytosolic Protein	3,004
	Secreted Protein	1,605
	Mitochondiral Protein	1,299
	Nuclear Protein	3,314
	Blinding	20

whole training and validation set is split to 80% training set and 20% validation set.

2.2. Data preprocessing and Features Engineering

To achieve the goal of prediction, data preprocessing is an important step. The first step is cleansing data by removing unknown amino acid represented X letter in protein sequence and replacing ambiguous code to the same format such as Asparagine (“B” or “N”) to “N”. The second process is feature engineering which creates features from the protein sequence. There are three main features categories : amino acid composition, physicochemical properties of amino acid and Nucleus localization signal (NLS).

The first feature category is amino acid composition features. This category contains four different measures which are length of amino acid sequence, a number of 20 amino acid in a protein, percentage of 20 amino acids in a protein and number of 400 dipeptide in a protein (Gao and et al, 2005), as shown in Table 2 below.

The second feature category is physicochemical properties of amino acid which is molecular weight, aromaticity, instability index, gravy, isoelectric point and secondary structure (helix, turn and sheet).

Molecular weight is the weight of amino acid sequence.

Aromaticity value is the relative frequency of Phenylalanine, Tryptophan, Tyrosine by using Lobry method, 1994. An aromatic amino acid is an amino acid that contain aromatic rings. These amino acids have ability to absorb light.

Instability index is the estimated value of stability of the amino acid sequence in a test tube. These method calculating by sum over the estimate weight value of each dipeptide from Guruprasad and et al in the amino acid sequence (Expasy).

Gravy is an average of hydropathy value for protein sequence by using Kyte and Doolittle technique (Expasy). This feature compress the hydrophilic and hydrophobic properties of each amino acid (Kyte and Doolittle, 1982).

Isoelectric point is pH value that make total of pH of amino acid equal zero.

Secondary structure percentage (helix, turn and sheet) is the percentage of helix, turn and sheet in amino acid sequence.

The last features category is Nucleus localization signal (NLS) flag. NLS is researched amino acid patterns that appear in a transported protein between in cell nucleus and out cell nucleus. To derive this feature, we check input sequence with NLS database from NLSdb (Bernhofer and et al., 2018).

Moreover, not only applying these feature categories in the whole sequence of input data but also applying protein composition features and physicochemical properties features in the first and the last 50 amino acid of input sequence to capture some pattern in the beginning or the end of amino acid sequences.

Table 2 : The first row of the table presents 20 amino acid code letters. Other rows present 400 dipeptide compositions.

G	**A**	**L**	**M**	**F**	**W**	**K**	**Q**	**E**	**S**	**P**	**V**	**I**	**C**	**Y**	**H**	**R**	**N**	**D**	**T**
GG	GA	GL	GM	GF	GW	GK	GQ	GE	GS	GP	GV	GI	GC	GY	GH	GR	GN	GD	GT
AG	AA	AL	AM	AF	AW	AK	AQ	AE	AS	AP	AV	AI	AC	AY	AH	AR	AN	AD	AT
LG	LA	LL	LM	LF	LW	LK	LQ	LE	LS	LP	LV	LI	LC	LY	LH	LR	LN	LD	LT
MG	MA	ML	MM	MF	MW	MK	MQ	ME	MS	MP	MV	MI	MC	MY	MH	MR	MN	MD	MT
FG	FA	FL	FM	FF	FW	FK	FQ	FE	FS	FP	FV	FI	FC	FY	FH	FR	FN	FD	FT
WG	WA	WL	WM	WF	WW	WK	WQ	WE	WS	WP	WV	WI	WC	WY	WH	WR	WN	WD	WT
KG	KA	KL	KM	KF	KW	KK	KQ	KE	KS	KP	KV	KI	KC	KY	KH	KR	KN	KD	KT
QG	QA	QL	QM	QF	QW	QK	QQ	QE	QS	QP	QV	QI	QC	QY	QH	QR	QN	QD	QT
EG	EA	EL	EM	EF	EW	EK	EQ	EE	ES	EP	EV	EI	EC	EY	EH	ER	EN	ED	ET
SG	SA	SL	SM	SF	SW	SK	SQ	SE	SS	SP	SV	SI	SC	SY	SH	SR	SN	SD	ST
PG	PA	PL	PM	PF	PW	PK	PQ	PE	PS	PP	PV	PI	PC	PY	PH	PR	PN	PD	PT
VG	VA	VL	VM	VF	VW	VK	VQ	VE	VS	VP	VV	VI	VC	VY	VH	VR	VN	VD	VT
IG	IA	IL	IM	IF	IW	IK	IQ	IE	IS	IP	IV	II	IC	IY	IH	IR	IN	ID	IT
CG	CA	CL	CM	CF	CW	CK	CQ	CE	CS	CP	CV	CI	CC	CY	CH	CR	CN	CD	CT
YG	YA	YL	YM	YF	YW	YK	YQ	YE	YS	YP	YV	YI	YC	YY	YH	YR	YN	YD	YT
HG	HA	HL	HM	HF	HW	HK	HQ	HE	HS	HP	HV	HI	HC	HY	HH	HR	HN	HD	HT
RG	RA	RL	RM	RF	RW	RK	RQ	RE	RS	RP	RV	RI	RC	RY	RH	RR	RN	RD	RT
NG	NA	NL	NM	NF	NW	NK	NQ	NE	NS	NP	NV	NI	NC	NY	NH	NR	NN	ND	NT
DG	DA	DL	DM	DF	DW	DK	DQ	DE	DS	DP	DV	DI	DC	DY	DH	DR	DN	DD	DT
TG	TA	TL	TM	TF	TW	TK	TQ	TE	TS	TP	TV	TI	TC	TY	TH	TR	TN	TD	TT

2.3. XGBoost algorithm

In high development in machine learning techniques, one of the most successful algorithm is XGBoost. XGBoost was introduced by Tianqi Chen and Carlos Guestrin in 2014 (Chen and Gyestrin, 2016). This algorithm becomes the baseline model of machine learning competition in term of structured data. XGBoost is developed from boosted tree algorithm which solving the problem of limited computations resources. As XGBoost concept based on ensemble tree model, the model select the best feature in each step to improve the ability of classifying each target output. As similar to classic machine learning algorithm, objective function is applied. The objective function is a function that evaluates the model performance. This function is a combination between training loss and regularization which regularization help the model avoid overfitting problem.

$$obj(\theta) = L(\theta) + \Omega(\theta)$$

To improve the model performance, combining several tree models are necessary. As a result of it, the objective function is shown below.

$$obj(\theta) = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

The difference between XGBoost and random forest is XGBoost uses gradient descent to optimize the objective function. Therefore, the scaling features are not necessary for XGBoost. According to the method that model selects feature to split the data, scaling feature might not increase the performance of the model. Moreover, handling missing value also is the benefit of the algorithm. The model can automate the way of handling missing data.

2.4. Regularization and Early stopping

To avoid common problem of supervise learning, overfitting, regularization method and early stopping are applied. In this paper, Ridge Regression or L2 is selected to be a regularization to reduce overfitting problem. L2 regularization adds penalty value which is squared magnitude of coefficient to the loss function.

$$\Omega(\theta) = \lambda \sum_{j}^{p} \beta_j^2$$

λ is a parameter that control the effect of regularization. If λ is nearly 0, the regularization will not affect to objective function. On the other hand, if λ becomes nearly 1, it might increase the chance of underfitting problem.

Table 3 : presents the hyperparameter which are tuned by random search

Hyperparameter	Random values
Max depth	[5, 7, 10, 11, 12, **13**, 14, 15]
Learning rate	[0.001,0.01, 0.05, **0.1**]
Subsample	[**0.7**, 0.8, 0.9]
Column Sample by tree	[0.4, 0.5, 0.6, 0.7, 0.8, **0.9**]
Column Sample by level	[0.4, 0.5, **0.6**, 0.7, 0.8, 0.9]
Min Child weight	[0.5, 1.0, 3.0, 5.0, 7.0, **10.0**, 13.0]
Gamma	[0,0.1,0.2, 0.25, **0.5**]
Number of trees	[70, 100,400, **500**, 1,000]

Another technique that reduces the overfitting problem is early stopping. This technique is simply and effective in term of reducing overfitting by stopping to train the model when objective function has not improve in n-steps over the training steps.

2.5. Random Search Algorithm

Tuning hyper-parameter is an important task to achieve the goal of machine learning development. The classic technique which is widely used is Grid Search. Grid Search algorithm creates a set of all parameters and finds the best set which is used to train the final model. This method might provide the best parameter set but it is also expensive in term of time consuming. In contrast, Random Search algorithm picks a set

Table 4 : compares the accuracy between experiments.

Input Features	Accuracy on Test set
Amino acid feature category	68.34%
Amino acid feature category and Physicochemical	69.26%
Amino acid feature category, Physicochemical and NLS flag	89.26%

of hyper-parameter randomly which method might get the less effective combination set but it save a large number of time that is spent to tune the model.

3. Experiment

XGBoost is an algorithm that has several hyper parameters to tune. The list of hyper parameter is shown in Table 3. To solved this task, Random search cross validation is applied to find the best hyperparameter combination. The features set which uses all of derived features that is used to do the experiment come from the Random Search from five cross validation. The best feature set that provides the highest score is bolded and shown in Table 3. To avoid overfitting problem, L2 regularization and 10 steps early stopping is set.

The first experiment is using the amino acid composition features. The second experiment is adding physicochemical properties to the first experiment. The last experiment using full derived features. The result of these experiment shown in Table 4 below.

Table 5 shows the accuracy in each subcellular location

Subcellular Location	**Amino acid feature category and Physicochemical**	**Fully model Accuracy**
Cytosolic Protein	59.39%	84.00%
Mitochondiral Protein	73.09%	77.02%
Nuclear Protein	67.06%	98.19%
Secreted Protein	88.14%	89.90%

Fig 1 shows the confusion matrix of the final model on test

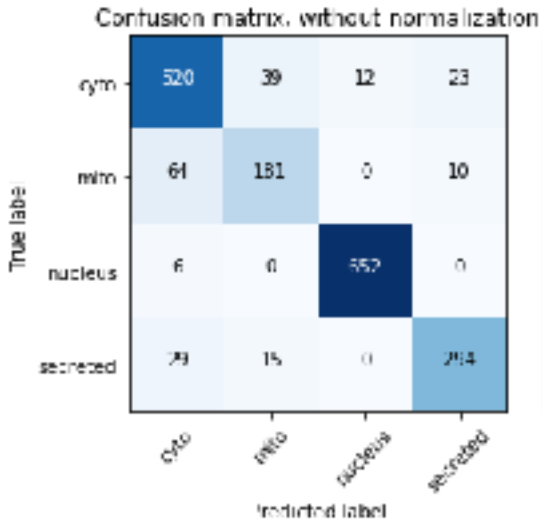
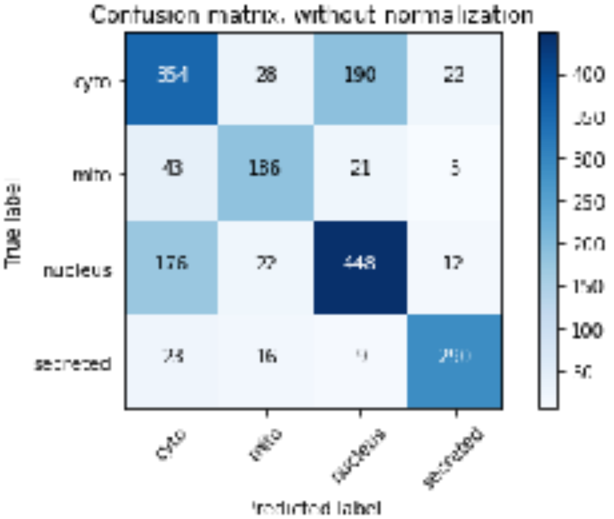


Fig 2 shows the confusion matrix of the second experiment model on test set.



4. Result and Discussion

As a result of the experiments, training XGBoost algorithm with all of the features is the highest accuracy, 89.26% on test set. The accuracy in each proteins is shown in Table 5. Confusion matrix is shown in Figure 1. Precision, Recall and F1 score are shown in Table 6. Top 10 features importance is shown in Table 7. Top 10 features importance from the final model is shown in Table 8.

Table 6 shows the top 10 features importance from the model.

Features	**Feature Importance**
Average hydropathy of the first 50 amino acid sequence	0.024629
NLS flag	0.019024
Isoelectric point of the first 50 amino acid sequence	0.018709
Isoelectric point	0.015867
Percent of Cysteine ©	0.014288
Instability index	0.013656
Percent of Leucine (L)	0.012788
Percent of helix in secondary structure	0.012393
Percent of sheet in secondary structure	0.011999
Percent of Glycine (G)	0.011288

Table 7 shows the precision, recall and F1 score in each subcellular location

Subcellular Location	Precision	Recalll	F1 Score
Cytosolic Protein	0.84	0.88	0.86
Mitochondiral Protein	0.77	0.71	0.74
Nuclear Protein	0.98	0.99	0.99
Secreted Protein	0.90	0.87	0.88

The final model shows significant improvement after adding NLS flag which increases total accuracy from 69.26% to 89.26%. Interestingly, the accuracy of Nuclear Protein subcellular location classification increases from 67.06% to 98.19% , as shown in Table 5, which might be caused by NLS flag. Feature importance measure also shows that NLS flag is the second informative feature for training model. It might be assumed that NLS flag help model to classify the difference between Cytosolic protein and Nuclear protein. The NLS patterns might appear in a large number of nuclear protein sequence. Not only NLS flag improves the accuracy of Nuclear protein classification, but also NLS flag considerably improves the accuracy of Cytosolic protein classification. The Cytosolic protein increases from 59.39% to 84.00%. Another evidence appears between confusion matrix of both models, as shown in Figure 1 and Figure 2. The confusion matrix of second experiment model on test set shows that the model misclassifies the target between Cytosolic protein and Nuclear protein 190 and 176, respectively. After adding NLS flag, the misclassification between Cytosolic protein and Nuclear protein reduce to 12 and 6.

Top 10 feature importance, as shown in Table 6, present the top 10 effective features which could be considered. Two of top three in the rank come from the first 50 amino acid sequence input which might assume that there are some considerable patterns appear in the beginning of amino acid sequence. However, the dipeptide features do not have significant impact on the model.

Table 8 shows the prediction of the model on blind test.

	Predict	Confident
SEQ677	secreted	60.76%
SEQ231	secreted	93.49%
SEQ871	secreted	58.54%
SEQ388	cyto	90.23%
SEQ122	cyto	76.70%
SEQ758	cyto	99.17%
SEQ333	cyto	80.21%
SEQ937	cyto	92.71%
SEQ351	cyto	87.30%
SEQ202	mito	80.51%
SEQ608	mito	78.61%
SEQ402	mito	85.71%
SEQ433	secreted	77.29%
SEQ821	secreted	89.82%
SEQ322	cyto	83.43%
SEQ982	cyto	82.53%
SEQ951	cyto	94.10%
SEQ173	cyto	98.79%
SEQ862	mito	58.39%
SEQ224	cyto	77.09%

To analysis error from classification, precision, recall and F1 score are used, as shown in Table 7. The overall of F1 score in the model is around 0.8675 which reflects the performance of the model. The highest F1 score is 0.99 which is Nuclear protein. The lowest F1 score is 0.74 which is Mitrochondrial protein.

Finally, Table 8 presents the classification result on the blind data set.

5. Conclusions

As results of experiments, it would be presented that model can achieve well which is in term of classification su To make classification model on subcellur location of eukaryotes protein by using the machine learning techniques (XGBoost), there are several categories which might improve model performance. According to experiments, amino acid composition and physicochemical properties, especially average hydropathy and isoelectric point are the main feature which help model classify the difference between Mitochondrial protein and Secreted protein. However, there feature categories are not enough to separate the difference between Cytosolic protein and Nuclear protein. The NLS flag feature comes to solve this problem. The result from the experiments presents that NLS flag is significant feature to train the model learning the difference between these protein.

6. References

Bernhofer M and et al. 2018, *NLSdb—major update for database of nuclear localization signals and nuclear export signals*, *Nucleic Acids Research*, Volume 46, Issue D1, Pages D503–D508, Available at <https://doi.org/10.1093/nar/gkx1021> [Accessed 19 March 2018]

Chen T and Gyestrin C, 2016. XGBoost: A Scalable Tree Boosting System. Available at <https://arxiv.org/abs/1603.02754> [Accessed 20 March 2018]

Expasy, *ProtParam* Available at <https://web.expasy.org/protparam/protparam-doc.html> [Accessed 22 March 2018]

Gao Q and et al, 2005. *Prediction of protein subcellular location using a combined feature of sequence. FEBS Letters,* Volume 579, Issue 16.

Kyte J and Doolittle R, 1982. *A simple method for displaying the hydropathic character of a protein,* Journal of Molecular Volume 157, Issue 1, Pages 105-132. Available at <https://www.sciencedirect.com/science/article/pii/0022283682905150?via%3Dihub> [Accessed 23 March 2018

7. Appendix

The code of this paper is available via the link below.
<https://github.com/zczlhaw/Bioinformatic_CW>

The code separates in to 5 sections.

1. Importing packages
2. Loading and preprocessing data
3. Training and Tuning model
4. Evaluating model
5. Predicting Blinding dataset