http://www.scirp.org/journal/jbise

J. Biomedical Science and Engineering, 2018, Vol. 11, (No. 6), pp: 126-143

Improving Protein Sequence Classification Performance Using Adjacent and Overlapped Segments on Existing Protein Descriptors

Mohammad Reza Faisal[1], Bahriddin Abapihi[1], Ngoc Giang Nguyen[1], Bedy Purnama[1], Mera Kartika Delimayanti[1], Dau Phan[1], Favorisen Rosyking Lumbanraja[1], Mamoru Kubo[2], Kenji Satou[2]

[1]Graduate School of Natural Science and Technology, Kanazawa University, Kanazawa, Japan; [2]Institute of Science and Engineering, Kanazawa University, Kanazawa, Japan

Correspondence to: Mohammad Reza Faisal, Reza.faisal@gmail.com

Keywords: Protein Sequence Classification, Protein Descriptor, Sequence Segmentation, Feature Selection

Received: May 28, 2018 **Accepted:** June 26, 2018 **Published:** June 29, 2018

ABSTRACT

In protein sequence classification research, it is popular to convert a variable length sequence of protein into a fixed length numerical vector by using various descriptors, for instance, composition of k-mer composition. Such position-independent descriptors are useful since they are applicable to any length of sequence; however, positional information of subsequence is discarded even though it might have high contribution to classification performance. To solve this problem, we divided the original sequence into some segments, and then calculated the numerical features for them. It enables us to partially introduce positional information (for instance, compositions of serine in anterior and posterior segments of a sequence). Through comprehensive experiments on the number of segments and length of overlapping region, we found our classification approach with sequence segmentation and feature selection is effective to improve the performance. We evaluated our approach on three protein classification problems and achieved significant improvement in all cases which have a dataset with sufficient amino acid in each sequence. This result has shown the great potential of using additional segments in protein sequence classification to solve other sequence problems in bioinformatics.

1. INTRODUCTION

Protein sequence is an essential asset in protein classification research. To apply different machine learning approaches on protein sequence data, it is a standard process to convert protein sequence into a

numerical representation. This process is called feature extraction and it is a critical step because the selection of the effective and appropriate type of feature extraction will profoundly affect classification performance. It drives the scientists to develop algorithm or program that performs feature extraction process, which is commonly known as protein descriptors [1].

Within two decades, scientists have developed various protein descriptors. Moreover, they have used these descriptors for various cases of protein analysis. Xiao *et al.* [1] grouped the types of commonly used descriptors into eight groups such as Amino Acid Composition, Autocorrelation, CTD, Conjoint Triad, Quasi-Sequence-Order, Pseudo-Amino Acid Composition, Proteochemometric descriptors, and PSSM. These groups have 22 type descriptors that have been actively used in researches.

The following are the commonly used protein descriptors and their application in protein analysis researches. Bhasin and Gajendra [2] used Amino Acid Composition (AAC) and Dipeptide Composition (DC) in their study to predict nuclear receptor. They used Support Vector Machine (SVM) as a classifier and achieved overall accuracy 82.6% when using numerical features from AAC and 97.5% with DC. The study about the prediction of membrane protein types was carried out by Feng and Zhang [3]. They adopted a formulation of the autocorrelation functions based on the hydrophobicity index of the 20 amino acids as protein descriptor. Using Bayes discriminant algorithm as a classifier, they achieved overall predictive accuracy as high as 94% and 82% for the re-substitution and jackknife tests. This result is higher about 13% in the resubstitution test and 8% in the jackknife test if compared with those of algorithms based only on the amino acid composition. Dubchak *et al.* [4] conducted a study on protein folding prediction using the global description of amino acid sequences or also known as CTD (Composition/Translation/Distribution) as protein descriptor. Using a neural network as a classifier, they obtained 71.7% accuracy for positive class prediction and 90% - 95% for negative class. In 2007, Shen *et al.* [5] presented a computational approach for predicting protein-protein interaction (PPI).The Support Vector Machine (SVM) algorithm was used to develop the methodology. They constructed numerical features for representing the PPI information by using conjoint triad descriptor. On average, their method may produce a PPI prediction model with an accuracy of 83.90% ± 1.29%. Another commonly used protein descriptor is quasi-sequence order descriptor. This descriptor was used by Chou [6] to solve prediction of protein subcellular locations. The author used this descriptor and augmented covariant discriminant algorithm as a classifier, and achieved accuracy between 79.6% - 86.4%.

Amino Acid Composition (AAC) is one of the protein descriptors often used to solve many cases of protein analysis. AAC has information from 20 amino acid components but does not have positional (*i.e.* sequence order) information. To increase the descriptor's ability, Chou [7] developed Pseudo Amino Acid Composition (PseAAC) by adding a set of sequence correlation factors. Ussing the PseAAC, a significant improvement in protein subcellular location prediction quality has been inspected for both the ProtLock algorithm and the covariant discriminant algorithm. In another study, the author combined 20 features from amino acid composition and 2λ numbers of a set of correlation factors that reflected different hydrophobicity and hydrophilicity distribution patterns along a protein chain [8]. Moreover, it also achieved better performance on the prediction of 16 subfamily classes of oxidoreductases if compared with AAC.

Protein descriptors described above can be grouped as an alignment-free descriptor. Also, there are descriptors grouped as alignment-based descriptor [9] or profile-based descriptor [10]. Profile-based descriptor generates feature vector based on Position-Specific Scoring Matrix (PSSM) by running PSI-BLAST. It produces some feature vectors that vary according to the amount of amino acid in the sequence. Rangwala and Karypis [11] used this descriptor to solve detection of remote homology and fold recognition. It can improve the overall ability to recognize remote homologs and distinguish proteins that share the same structural fold.

Existing protein descriptors perform feature extraction using information such as hydrophobicity, polarizability, polarity, charge, surface tension, secondary structure, solvent accessibility and normalized Van der Waals volume. However, Asgari and Mofrad [12] developed a protein descriptor without that information. They adopted existing methods in natural language processing (NLP) that is Continuous Vector Representation, as a distributed representation for words. Testing was performed on 7027 protein fam-

ilies using SVM as a classifier. They obtained a weighted average accuracy of 93% ± 0.06%.

A combination of several existing descriptors can also generate a new numerical representation of protein sequence. This numerical representation has more information than the features generated only from a descriptor, and it can improve prediction accuracy. This study was carried out by Ong *et al.* [13] in 2007 for predicting protein functional families. They used various descriptors of an alignment-free group such as Amino Acid Composition, Dipeptide Composition, Normalized Moreau-Broto Autocorrelation, Moran Autocorrelation, Geary Autocorrelation, Quasi Sequence Order, Pseudo Amino Acid Composition, and Descriptors of Composition, Transition, and Distribution. They gained a slightly better prediction performance than the use of individual descriptor. In other research, Liu *et al.* [14] conducted a study of alignment-free and alignment-based descriptor combinations using Pseudo Amino Acid Composition (PseAAC) and Profile-based descriptor. They proposed two methods to solve the remote protein homology detection. The first method, named PseAAC Index, is a combination of features from PseAAC and 531 indices extracted from the AA Index database. This method can get the average ROC score 0.88. The second method is a combination of PseAAC Index with a profile-based protein representation. They are named PseAAC Index-Profile, which obtained the average ROC score 0.922. From these researches, the combination of features from various protein descriptors can improve prediction performance in general. However, according to a study by Ong *et al.* [13], those features may not always improve prediction performance because they contain noises. The authors suggested the use of feature selection method to reduce noises and choose important features.

One common thing in these researches is that only a full length of the sequence is used as an input to the protein descriptor. It means that the output of the protein descriptor only describes the state of a whole protein alone. If the descriptor has a segment of the sequence as an input, it will give information of that segment. We can expect amixture of numeric representation from a full length of the sequence and its segments provide information of the whole protein state (global information) as well as the state of each segment (local information). In this study, we propose an effective approach for improving existing alignment-free protein descriptor capabilities by using adjacent and overlapped segments as inputs. We also tried to use a combination of various descriptors with this input. With this approach, we have improved prediction performance in several validation datasets. However, this approach may have features with noises generated through the use of overlapping and redundant descriptors. Accordingly, by exploiting feature selection along feature ranking, we achieved slight improvements in prediction accuracy, and at the same time, we could also find which type of features was more useful to increase it.

The remainder of this paper is organized as follows. In Section 2, we present in detail about how the model works. In Section 3, we show experiments and results of evaluating the model with some validation datasets. Finally, some discussions and conclusions are given in Section 4.

2. METHODS

2.1. Existing Alignment-Free Protein Descriptors

In this research, we used the protein descriptor from R package protr. This package has various structures and physicochemical descriptors and PCMs modeling descriptors for amino acid sequence [1]. A list of protein descriptors covered by protr is presented in **Table 1**.

Protr has eight group descriptors. The first seven groups are the alignment-free descriptors and the last group, PSSM, is an alignment-based descriptor. The PSSM group has PSSM profile descriptor that produces outputs with a varying number of features depends on the number of amino acid.

In active research on protein classification, feature extraction is one of the important processes. This process converts a protein sequence into numerical features by using protein descriptor. If s is a protein sequence with n amino acids, where $s_i \in \{\mathrm{A},\mathrm{R},\mathrm{N},\mathrm{D},\mathrm{C},\mathrm{E},\mathrm{Q},\mathrm{M},\mathrm{F},\mathrm{P},\mathrm{S},\mathrm{T},\mathrm{W},\mathrm{Y},\mathrm{V}\}$.

$$s_1\ s_2\ s_3\ \cdots\ s_n$$

The protein descriptor can then be written as the following formula:

Table 1. Description of existing protein descriptors.

No.	Descriptor	Group	# Features
1	Amino acid composition	Amino acid composition	20
2	Dipeptide composition		400
3	Tripeptide composition		8000
4	Normalized Moreau-Broto	Autocorrelation	240[a]
5	Moran		240[a]
6	Geary		240[a]
7	Composition	CTD	21
8	Transition		21
9	Distribution		105
10	Conjoint Triad	Conjoint Triad	343
11	Sequence-order-coupling number	Quasi-sequence-order	60[a]
12	Quasi-sequence-order descriptors		100[a]
13	Type I	Pseudo-amino acid composition	50
14	Type II		80
15	Principal components analysis (amino acid properties based)	Proteochemometric descriptors	175[b]
16	Principal components analysis (2D and 3D molecular descriptors based)		4025[b]
17	Factor analysis (amino acid properties based)		175[b]
18	Factor analysis (2D and 3D molecular descriptors based)		4025[b]
19	Multidimensional scaling (amino acid properties based)		175[b]
20	Multidimensional scaling (2D and 3D molecular descriptors based)		4025[b]
21	BLOSUM and PAM matrix-derived descriptors		175[b]
22	PSSM profile	PSSM	-

[a]The number of descriptor's features output depends on the selection of the number of properties of amino acid and the selection of the parameter. [b]The number of descriptor's features output depends on the selection of the number of components and the selection of the lag parameter.

$$descriptor(s) = f \tag{1}$$

The output of $descriptor(s)$ is numerical features f where $f_j \in$ decimal numbers and m is the number of features.

$$f_1, f_2, f_3, \cdots, f_m$$

The use of a single protein descriptor based classifier has solved protein analysis cases. It predicts nuclear receptor [2], membrane protein types [3], protein folding [4], protein-protein interaction (PPI) [5], and protein subcellular locations [6, 8]. It also detects the remote homology and folds recognition [11].

To obtain more sequence's information and to improve prediction accuracy, acombination of various descriptors is also used to generate anumerical representation of protein sequence in general active research. This formula can represent a combination of various descriptors implementation:

$$\bigcup_{type} descriptor_{type}(s) = \bigcup_{type} f_{type} \tag{2}$$

where $type$ is descriptor type, $type \in$ {amino acid composition, dipeptide composition, tripeptide composition, and other descriptors that listed in **Table 1**}.

f_{type} is numerical features, $f_{type,1}$, $f_{type,2}$, ···, $f_{type,m}$ where $f_{type,j} \in$ integer, $j = 1, 2, \cdots, m$ and m is the number of features which are generated by $descriptor_{type}$. For instance, if we use two type of descriptors such as Amino Acid Composition (aac) and Dipeptide Composition (dt) then we have numerical features as shown below.

$$descriptor_{aac}(s) \cup descriptor_{dc}(s) = f_{aac} \cup f_{dc} = f_{aac,1}, \cdots, f_{aac,20}, f_{dc,1}, \cdots, f_{dc,400}$$

One of the successful reports of this approach is the study of predicting protein functional families by using a combination of eight descriptors from alignment-free groups [13]. Moreover, the other study used a combination of alignment-free descriptors and alignment-based descriptors for remote protein homology detection [14]. Both of that studies had same conclusion that the combination of various descriptors can give a better result than using a single descriptor only.

2.2. Protein's Features Construction

Equations (1) and (2) can represent the feature extraction process that has been used in active research. One common thing in both equations is that they use a full-length of sequence s as the input. Moreover, f is the output which provides global information of s.

Our goal is to construct protein's features that have complete information, not only global information but also local information. If the sequence s is divided into several segments, and each segment becomes input to a descriptor. Then each output has local information on its location. We obtained new features by concatenation all those outputs. The division of those segments is done in two steps.

In the first step, we generated segments that have relatively same length. The first segment is calculated from the beginning of the sequence, then followed by the second segment and so on. We named this segment as adjacent segment. For example, given a protein sequence s as shown below:

MCMDVRCPSICTAPGSRGLASACMERVCIC

If we divide sequence s into k segments where $k = 3$, then the generated segments are as follows:

$segment_1$ = MCMDVRCPSI

$segment_2$ = CTAPGSRGLA

$segment_3$ = SACMERVCIC

With the following formula where $n_{segment}$ is a initial number of amino acids in each segment:

$$n_{segment} = \lceil n_s / k \rceil \tag{3}$$

where n_s is a number of amino acids in sequence s. Each segment is then generated as follows:

$$segment_j = s_{start}\ s_{start+m}\ \cdots\ s_{end} \tag{4}$$

$$start = (j-1) * n_{segment} + 1 \tag{5}$$

$$end = j * n_{segment} \tag{6}$$

and for the last segment when $k = j$:

$$end = n_{sequence} \tag{7}$$

where $1 \le m \le (end - start)$ and $1 \le j \le k$.

In the second step, we generate additional segments to get local information between two adjacent segments. We named this segment as overlapped segment. An overlapped segment is the union of half from the end of the first segment and a half from the beginning of the second segment. For example, an overlapped segment for $segment_1$ and $segment_2$ is obtained as follows:

$$overlapped_1 = \frac{1}{2}segment_1 \cup \frac{1}{2}segment_2 = MCMDVRCPSI \cup CTAPGSRGLA = RCPSICTAPG$$

$$overlapped_2 = \frac{1}{2}segment_2 \cup \frac{1}{2}segment_3 = CTAPGSRGLA \cup SACMERVCIC = SRGLASACME$$

Each overlapped segment can be generated using the following formula:

$$overlapped_l = \frac{1}{2}segment_l \cup \frac{1}{2}segment_{l+1} \tag{8}$$

where $1 \le l \le (k-1)$.We generate amino acids of $\frac{1}{2}segment_l$ with following formula:

$$\frac{1}{2}segment_l = s_{start}\ s_{start+m}\ \cdots\ s_{end} \tag{9}$$

$$start = \left((j-1) * n_{segment} + 1\right) + \frac{1}{2} * n_{segment} \tag{10}$$

$$end = j * n_{segment} \tag{11}$$

where $1 \le m \le (end - start)$ and $1 \le j \le k$. And $\frac{1}{2}segment_{l+1}$ is generated by using formula below:

$$\frac{1}{2}segment_{l+1} = s_{start}\ s_{start+m}\ s_{end} \tag{12}$$

$$start = (j-1) * n_{segment} + 1 \tag{13}$$

$$end = j * \frac{1}{2} * n_{segment} \tag{14}$$

After all segments are created, we calculate features of sequences s by using the formulabelow:

$$descriptor(s) \cup \left(\bigcup_{i=1}^{k} descriptor(segment_i)\right) \cup \left(\bigcup_{l=1}^{k-1} descriptor(overlapped_l)\right) \tag{15}$$

The result of the above formula is numerical features as defined below:

$$f_s \cup \bigcup_{i=1}^{k} f_{segment_i} \cup \bigcup_{l=1}^{k-1} f_{overlapped_l} \tag{16}$$

For instance, if sequence s is divided into k segments ($k = 3$) and protein descriptor is Amino Acid

Composition. Accordingly, the generated features are:

$$f_s = f_1, \cdots, f_{20}$$

$$\bigcup_{i=1}^{k} f_{segment_i} = f_{segment_{1,1}}, \cdots, f_{segment_{1,20}} \cup f_{segment_{121}}, \cdots, f_{segment_{2,20}} \cup f_{segment_{3,1}}, \cdots, f_{segment_{3,20}}$$

$$\bigcup_{l=1}^{k-1} f_{overlapped_l} = f_{overlapped_{1,1}}, \cdots, f_{overlapped_{1,20}} \cup f_{overlapped_{2,1}}, \cdots, f_{overlapped_{2,20}}$$

By using $k = 3$, the numerical representation of sequence s has 120 numerical features.

In our study, we expect that the use of various values of k will provide complete information of sequence s than the use of single k value. For example $k = 2, 3, \cdots, z$, where z is a positive integer. Moreover, we can generate numerical features for sequence s as defined below:

$$descriptor(s) \cup \bigcup_{k=2}^{z} \left(\left(\bigcup_{i=1}^{k} descriptor(segment_i) \right) \cup \left(\bigcup_{l=1}^{k-1} descriptor(overlapped_l) \right) \right) \quad (17)$$

We also implement this approach with a combination of various descriptors. So the sequence s will have numerical features as follows:

$$\bigcup_{type} \left(descriptor_{type}(s) \cup \bigcup_{k=2}^{z} \left(\left(\bigcup_{i=1}^{k} descriptor_{type}(segment_i) \right) \cup \left(\bigcup_{l=1}^{k-1} descriptor_{type}(overlapped_l) \right) \right) \right) \quad (18)$$

2.3. Algorithm

Our proposed approach consists of main three steps. The flowchart of our approach is shown in Figure 1.

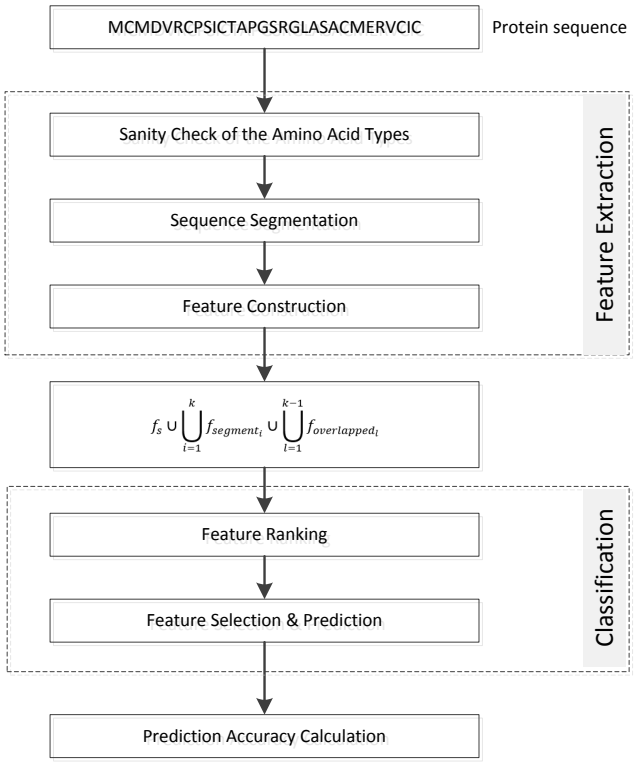

Figure 1. The flowchart of the proposed approach.

The first step is feature extraction that has three processes:

1) Sanity check of the amino acid types is responsible for erasing amino acids if they are not in the 20 default of amino acid types.

2) Sequence segmentation is conducted for dividing a sequence into adjacent segments and overlapped segments.

3) Feature construction is in charge of converting a original sequence, adjacent segments, and overlapped segments into numerical features by using existing descriptor from protr package. Then a concatenation of all those numerical features is created.

The second step is classification. This step has two processes that are commonly used in active classification research. We conduct k-fold cross-validation or jackknife test, each process in this step are repeated k times or n time, with n is a number of samples.

1) Feature ranking is responsible for sorting features by importance. The random Forest function for R [15] conducts this process.

2) Feature selection and prediction are responsible for creating feature subsets, and performing learning and predicting with ksvm function in a kernlab package for R [16].

The last step is prediction accuracy calculation. It is in charge of calculating accuracy for each feature subset.

3. EXPERIMENTS AND RESULTS

In order to show the validity of our proposed approach to improving existing alignment-free protein descriptor to deal with a protein sequence classification problem, we did experiments with datasets from UniProt, Swiss-Prot, and Nuclea RDB. Our experiments are grouped by three protein analysis cases which are a classification of nuclear receptors, protein family classification, and cell-penetrating peptides prediction.

3.1. Classification of Nuclear Receptors

In this section, we evaluate the strength of our proposed approach on a single protein descriptor in the classification of nuclear receptors. Nuclear receptors are key transcription factors that regulate important gene networks responsible for cell growth, differentiation, and homeostasis [2]. Classification of nuclear receptors was done in researches [2, 17].

As done by Bhasin and Gajendra [2], the classification was achievedon the basis of amino acid composition and dipeptide composition from a sequence of nuclear receptors using support vector machine (SVM).They did training and testing on a non-redundant dataset of 282 proteins obtained from the NucleaRDB database. The dataset had four subfamilies of nuclear receptors as shown in **Table 2**.

The performance of both classifiers was evaluated using 5-foldcross-validation.The accuracy of the amino acid composition-based classifier was 82%, and dipeptide composition-based classifier was 97.5%.

In the research done by Wang *et al.* [17], the classification was achieved on the basis of various protein descriptors from a sequence of nuclear receptors using Fuzzy K nearest neighbor (FK-NN).They

Table 2. Description of the datasetin Bhasin and Gajendra research.

No.	Nuclear receptor subfamilies	# sequence
1	NR1: Thyroid hormone-like	114
2	NR2: HNF-4-like	72
3	NR3: Estrogen-like	75
4	NR5: Fusi-tarazu-like	21

converted a sequence into numerical features by using a combination of amino acid composition, dipeptide composition, complexity factor and low-frequency Fourier spectrum components. The training and testing were done on 159 sequences of nuclear receptors obtained from NucleaRDB database and 500 sequences of non-nuclear receptors obtained from UniProt database. No sequence had ≥60% sequence identity with any other sequence in this dataset. Nuclear receptors data had seven subfamilies as shown in **Table 3**.

They create two layers predictor. The first layer was used to identify a query protein as NR or not. If it was a NR, the second layer would be continued to identify the NR among the seven subfamilies. The performance of all classifier was evaluated using jackknife test and independent dataset test. The overall accuracy of first layer predictor is 92.56% by using jackknife test and 98.03% by using independent dataset test. Moreover, the overall accuracy of second layer predictor is 88.68% by using jackknife test and 99.65% by using independent dataset test.

Research [2] is a single descriptor based classifier and research [17] can be grouped as various descriptors based classifier. Both researches have similarities. They use the same type of descriptor which are amino acid composition and dipeptide composition.

To compare the results of our proposed approach to the result of research [2], we used their method on the data those were provided by the research [17]. However, we use four subfamilies; they are the same subfamilies that were used in research [2] as shown in **Table 4**.

We also used same classifier and evaluation method which are Support Vector Machine with a 5-fold cross-validation test.

Table 3. Description of the datasetin Wang *et al.* research.

No.	Set	Subfamily	# sequence
1	Nuclear receptors (NR)	NR1: thyroid hormone like	50
2		NR2: HNF4-like	36
3		NR3: estrogen like	37
4		NR4: nerve growth factor IB-like	7
5		NR5: fushitarazu-F1 like	12
6		NR6: germ cell nuclear factor like	5
7		NR0: knirps and DAX like	12
8	Non-nuclear receptors (Non-NR)	N/A	500

Table 4. Description of the modifieddataset in our research.

No.	Nuclear receptor subfamilies	# sequence
1	NR1: Thyroid hormone-like	50
2	NR2: HNF-4-like	36
3	NR3: Estrogen-like	37
4	NR5: fushitarazu-F1 like	12

In this experiment, we converted a sequence into numerical features by using Equation (17). In amino acid composition based classifier experiment, we obtained the best prediction accuracy at z = 7. Moreover, in dipeptide composition based classifier experiment, the best prediction accuracy was achieved at z = 4. The comparison of our experimental results and result from methods from research [2] is shown in Table 5.

We also investigated important features that have contributed to the prediction accuracy. Table 6 and Table 7 show detail of 790 important features that were obtained in AAC_7 FS experiment and 355 important features that were generated in DC_4 FS experiment.

For further, we compared research [17] results with our result. In the experiment of identifying NR and non-NR, we used amino acid composition based classifier with z = 3 and dipeptide composition based classifier with z = 2. The result is shown in Table 8.

Detail important features of AAC_3 FS and DC_2 FS are shown in Table 9 and Table 10.

In the second level experiment, we identified NR subfamilies by using amino acid composition based classifier with z = 5 and dipeptide composition based classifier with z = 2. The comparison result is shown in Table 11. Moreover, the detail of important features on AAC_5 FS and DC_2 FS experiments are shown in Table 12 and Table 13.

Table 5. Prediction accuracy comparison of our approach and method in research [2].

No.	Method	Accuracy (%)	# Features	Description
1	AAC	67.99	20	AAC based classifier of Research [2].
2	DC	93.60	400	DC based classifier of Research [2].
3	AAC_7	86.97	980	AAC based classifier with z = 7.
4	DC_4	94.19	6400	DC based classifier with z = 4.
5	AAC_7 FS	88.06	790	AAC based classifier with z = 7 and feature selection.
6	DC_4 FS	**96.19**	355	DC based classifier with z = 4 and feature selection.

Table 6. Detail of important features in AAC_7 FS experiment.

Source	# Important Feature	# Total Features
Original sequence	14	20
k = 2	52	60
k = 3	79	100
k = 4	116	140
k = 5	141	180
k = 6	180	220
k = 7	208	260
Total	790	980

Table 7. Detail of important features inDC_4 FS experiment.

Source	# Important Feature	# Total Features
Original sequence	34	400
k = 2	90	1200
k = 3	124	2000
k = 4	107	2800
Total	355	6400

Table 8. Prediction accuracy comparisonof our approach and method in research [17] for identifying NR and non-NR.

No.	Method	Accuracy (%)	# Features	Description
1	NR-2L	92.56	881	Result by Wang *et al.*
2	AAC_3	97.56	180	AAC based classifier with z = 3
3	DC_2	97.87	1600	DC based classifier with z = 2
4	AAC_3 FS	97.87	100	AAC based classifier with z = 3 and feature selection
5	DC_2 FS	**98.48**	120	DC based classifier with z = 2 and feature selection

Table 9. Detail of important features inAAC_3 FS experiment.

Source	# Important Feature	# Total Features
Original sequence	11	20
k = 2	36	60
k = 3	53	100
Total	100	180

Table 10. Detail of important features inDC_2 FS experiment.

Source	# Important Feature	# Total Features
Original sequence	37	400
k = 2	83	1200
Total	120	1600

Table 11. **Prediction accuracy comparison of our approach and method in research [17] for identifying NR subfamilies.**

No.	Method	Accuracy (%)	# Features	Description
1	NR-2L	88.68	881	Result by Wang *et al.*
2	AAC_5	81.76	500	AAC based classifier with z = 5
3	DC_2	91.81	1600	DC based classifier with z = 2
4	AAC_5 FS	83.01	355	AAC based classifier with z = 5 and feature selection
5	DC_2 FS	**94.33**	145	DC based classifier with z = 2 and feature selection

Table 12. **Detail of important features inAAC_5 FS experiment.**

Source	# Important Feature	# Total Features
Original sequence	13	20
k = 2	42	60
k = 3	74	100
k = 4	96	140
k = 5	130	160
Total	355	480

Table 13. **Detail of important features of DC_2 FS experiment.**

Source	# Important Feature	# Total Features
Original sequence	43	400
k = 2	102	1200
Total	145	1600

3.2. Protein Family Classification

In this experiment, we evaluate the strength of our proposed approach on the combination of various protein descriptors. We selected protein family classification as the case. A protein family is a set of proteins that are evolutionarily related, typically involving similar structures or functions [12]. Protein family classification was done in researches [12, 18]. Cai *et al.* [18] had classified 54 functional families. The feature extraction process had been done by using a combination of protein descriptors which are composition, translation, and distribution. The reported accuracies of family classification had been in the range of 69.1% - 99.6%. In another study, Asgari and Mofrad [12] performed classifications of 7027 protein families. They applied a new feature extraction method as known as ProtVec. The average accuracy for the first 1000 families is 94% ± 0.05%.And the average accuracy for 2000, 3000 and 4000 frequent families were respectively 93% ± 0.05%, 92% ± 0.06%, and 91% ± 0.08%.The weighted accuracy of all 7027 families was

93% ± 0.06%.

In this experiment, we used the dataset that were provided by Asgari and Mofrad [12] and performed 1000 classification cases using the first 1000 families. The classification performed in this experiment is a balanced binary classification. Samples of positive class are samples of selected family protein. Samples of negative class are randomly selected samples. In the feature extraction process, we used a combination of various protein descriptors which are Amino Acid Composition (AAC), Composition (CTDC), translation (CTDT), and distribution (CTDD) with z = 5. Moreover, we used SVM with 10-fold cross-validation test as classifier and evaluation method. We used feature selection to check whether there was a significant increase in accuracy of prediction. There were improvements, but it was not significant as shown in **Table 14**.

We have investigated subset features that can obtain the best accuracy prediction from each family classification case. The result of our investigation of three families are shown in **Tables 15-17**. We saw a subset features were formed of the four descriptors that we used with all various k values.

Table 14. Prediction accuracy comparison of our approach and method in research [12] for classifying first 1000 families.

No.	Method	Description	Weighted Accuracy (%)
1	ProVec 1000	Asgari and Mofrad's method for the first 1000 families	93.95
2	Our Approach	Our method	96.19
3	Our Approach FS	Our method with feature selection	**96.79**

Table 15. Detail of important features in 50S ribosome-binding GTPase family classification.

Descriptor	Original	k = 2	k = 3	k = 4	k = 5	# total
AAC	13	36	53	64	84	250
CTDC	13	47	87	110	129	386
CTDT	11	35	55	79	98	278
CTDD	76	165	237	295	263	1036
						1950

Table 16. Detail of important features in Transmembrane receptor (rhodopsin family) family classification.

Descriptor	Original	k = 2	k = 3	k = 4	k = 5	# total
AAC	3	6	7	8	8	34
CTDC	8	24	33	35	34	134
CTDT	8	12	15	9	11	55
CTDD	7	8	3	4	5	27
						250

Table 17. **Detail of important features in Ribosomal protein S14p/S29e family classification.**

Descriptor	Original	k = 2	k = 3	k = 4	k = 5	# total
AAC	1	2	2	4	2	11
CTDC	3	6	7	3	3	22
CTDT	1	2	3	2	2	10
CTDD	2	2	2	0	1	7
						50

3.3. Cell-Penetrating Peptides Prediction

Cell-penetrating peptides (CPPs) are small peptides that are about 10 - 30 amino acids long. CPPs can carry various bioactive cargoes, ranging from small molecules to proteins and supramolecular particles, to directly enter cells without significantly damaging the cell membrane. It makes them potential drug delivery agents for the translocation of cargo into cells. CPP prediction research has increased in the past few years.CPPsite2.0 is CPP-specific database that has approximately 1850 experimentally validated CPPs [19].

CPPred-RF is one method that has succeeded to solve the CPPs prediction case [19]. In this study Wei *et al.* used two dataset that are CPP924 and CPPsite 3. The detail information of those dataset are shown in Table 18. In feature extraction process, they used a combination of several descriptors, *i.e.* parallel correlation pseudo-amino-acid composition (PC-PseAAC), series correlation pseudo-amino acid composition (SC-PseAAC), adaptive skip dipeptide composition (ASDC) and physicochemical properties (PPs).The result is numerical representation with 636 features. Then features selection is applied by using Max-Relevance-Max-Distance (MRMD) as feature ranking method and Sequential Feature Selection (SFS) as optimal features selector. Moreover, they used random forest as the classifier with jackknife test at the prediction and evaluation stage. The result is 91.6% Accuracy for CPP924 dataset and 71.1% accuracy for CPPsite3.

In this experiment, we implemented our approach on single descriptor and combination of various descriptors based classifier. We used amino acid composition, dipeptide composition and composition/distribution/translation (CTD) descriptor on feature extraction process. In the classification and evaluation process, we used SVM as a classifier with 10-fold cross-validation test. The results are shown in the tables (Table 19 and Table 20).

The best performance was obtained by using ACC based classifier with original input sequence. Implementation of our approach with z = 2 and z = 3 cannot produce better performance instead of decreasing accuracy. In the experiment a combination of various descriptors based classifier with those descriptors and feature selection, we obtained 76.08% accuracy and 20 important features.

4. DISCUSSIONS AND CONCLUSIONS

We have proven that our proposed approach is simple in implementation and powerful on solving protein sequence classification problems. Our approach was tested on three cases which are classification of nuclear receptors, protein family classification, and cell-penetrating peptides prediction. We compared the performance of our approach with the performance from other methods that have been used in those cases.

On first two classification cases, the experimental results show that there was a significant improvement in the prediction accuracy of our approach. We also used random Forest to generate variable importance to rank features, and then perform the feature ranking to conduct feature selection. Feature selection also helped us to get information that features subset which gave the best accuracy contains generated features from additional segments. Our approach also worked in both single descriptor and a combination

Table 18. **Dataset Description of the dataset inresearch [19].**

No.	Dataset	# positive	# negative	# amino acid
1	CPP924	462	462	10 - 61
2	CPPsite 3	187	187	5 - 61

Table 19. **The predictive result of the proposed approach on CPP924 dataset.**

No.	Descriptor	Source	Accuracy
		Original	**90.69**
1	Amino Acid Composition	z = 2	89.82
		z = 3	90.04
		Original	89.39
2	CTD-Composition	z = 2	88.31
		z = 3	88.74
		Original	85.06
3	CTD-Translation	z = 2	83.87
		z = 3	83.87
		Original	77.48
4	CTD-Distribution	z = 2	76.73
		z = 3	78.89
		Original	87.66
5	Dipeptide Composition	z = 2	87.55
		z = 3	84.30

various descriptors based classifier.

In contrast, our approach did not work well in Cell-Penetrating Peptides Prediction. Performance of our approach was not significantly improved, or it was lower than the result of the classifier with original sequence only. It occurred because sequences have a small number of amino acids. Table 21 shows the comparison of amino acids numbers from each case.

In this research, we only focus on solving protein sequence classification problems with five out of 21 of existing protein descriptors which are grouped to the alignment-free descriptor. In the future, we apply the proposed approach using other descriptors. Also, we need further investigation to find out the

Table 20. **The predictive result of the proposed approach on CPPsite 3dataset.**

No.	Descriptor	Source	Accuracy
1	Amino Acid Composition	Original	**64.97**
		z = 2	59.62
		z = 3	58.28
2	CTD-Composition	Original	63.36
		z = 2	58.02
		z = 3	58.82
3	CTD-Translation	Original	61.76
		z = 2	54.54
		z = 3	59.43
4	CTD-Distribution	Original	57.48
		z = 2	64.17
		z = 3	63.63
5	Dipeptide Composition	Original	62.03
		z = 2	60.96
		z = 3	64.20

Table 21. **Statistic comparison of amino acid numbers in sequences.**

No.	case	# amino acid				
		min	max	median	mean	mode
1	Classification of Nuclear Receptor	2	3932	419	510	419
2	Protein Family Classification	7	21531	332	425	101
3	Cell-Penetrating Peptides Prediction	5	61	17	19	18

minimum number of amino acid in sequence to make our approach can work properly.

ACKNOWLEDGEMENTS

In this research, the super-computing resource was provided by Human Genome Center, the Institute of Medical Science, the University of Tokyo. Additional computation time was provided by the super computer system in Research Organization of Information and Systems (ROIS), National Institute of Genetics (NIG). This work was supported by JSPS KAKENHI Grant Number 26330328.

REFERENCES

1. Xiao, N., Cao, D.-S., Zhu, M.-F. and Xu, Q.-S. (2015) protr/ProtrWeb: R Package and Web Server for Generat-

ing Various Numerical Representation Schemes of Protein Sequences. *Bioinformatics*, **31**, 1857-1859. https://doi.org/10.1093/bioinformatics/btv042

2. Bhasin, M. and Raghava, G.P.S. (2004) Classification of Nuclear Receptors Based on Amino Acid Composition and Dipeptide Composition. *The Journal of Biological Chemistry*, **279**, 23262-23266. https://doi.org/10.1074/jbc.M401932200
3. Feng, Z.-P. and Zhang, C.-T. (2000) Prediction of Membrane Protein Types Based on the Hydrophobic Index of Amino Acids. *Journal of Protein Chemistry*, **19**, 269-275. https://doi.org/10.1023/A:1007091128394
4. Dubchak, I., Muchnik, I., Holbrook, S.R. and Kim, S.H. (1995) Prediction of Protein Folding Class Using Global Description of Amino Acid Sequence. *Proceedings of the National Academy of Sciences of the USA*, **92**, 8700-8704. https://doi.org/10.1073/pnas.92.19.8700
5. Shen, J., Zhang, J., Luo, X., Zhu, W., Yu, K., Chen, K., Li, Y. and Jiang, H. (2007) Predicting Protein-Protein Interactions Based Only on Sequences Information. *Proceedings of the National Academy of Sciences of the USA*, **104**, 4337-4341. https://doi.org/10.1073/pnas.0607879104
6. Chou, K.-C. (2000) Prediction of Protein Subcellular Locations by Incorporating Quasi-Sequence-Order Effect. *Biochemical and Biophysical Research Communications*, **278**, 477-483. https://doi.org/10.1006/bbrc.2000.3815
7. Chou, K.-C. (2001) Prediction of Protein Cellular Attributes Using Pseudo-Amino Acid Composition. *Proteins*: *Structure*, *Function*, *and Bioinformatics*, **44**, 60. https://doi.org/10.1002/prot.1072
8. Chou, K.-C. (2005) Using Amphiphilic Pseudo Amino Acid Composition to Predict Enzyme Subfamily Classes. *Bioinformatics*, **21**, 10-19. https://doi.org/10.1093/bioinformatics/bth466
9. Phan, D., Nguyen, N.G., Lumbanraja, F.R., Faisal, M.R., Abapihi, B., Purnama, B., Delimayanti, M.K., Kubo, M., and Satou, K. (2017) Combined Use of k-Mer Numerical Features and Position-Specific Categorical Features in Fixed-Length DNA Sequence Classification. *Journal of Biomedical Science and Engineering*, **10**, 390-401. https://doi.org/10.4236/jbise.2017.108030
10. Xiao, N., Cao, D.-S., Zhu, M.-F. and Xu, Q.-S. (2017) protr: R Package for Generating Various Numerical Representation Schemes of Protein Sequences. https://cran.r-project.org/web/packages/protr/vignettes/protr.html
11. Rangwala, H. and Karypis, G. (2005) Profile-Based Direct Kernels for Remote Homology Detection and Fold Recognition. *Bioinformatics*, **21**, 4239-4247. https://doi.org/10.1093/bioinformatics/bti687
12. Asgari, E. and Mofrad, M.R.K. (2015) Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics. *PLoS One*, **10**, 1-11. https://doi.org/10.1371/journal.pone.0141287
13. Ong, S.A.K., Lin, H.H., Chen, Y.Z., Li, Z.R. and Cao, Z. (2007) Efficacy of Different Protein Descriptors in Predicting Protein Functional Families. *BMC Bioinformatics*, **8**, 300. https://doi.org/10.1186/1471-2105-8-300
14. Liu, B., Wang, X., Zou, Q., Dong, Q. and Chen, Q. (2013) Protein Remote Homology Detection by Combining Chou's Pseudo Amino Acid Composition and Profile-Based Protein Representation. *Molecular Informatics*, **32**, 775-782. https://doi.org/10.1002/minf.201300084
15. Liaw, A. and Wiener, M. (2002) Classification and Regression by randomForest. *R News*, **2**, 18-22.
16. Karatzoglou, A., Smola, A., Hornik, K. and Zeileis, A. (2004) kernlab-An {S4} Package for Kernel Methods in {R}. *Journal of Statistical Software*, **11**, 1-20. https://doi.org/10.18637/jss.v011.i09
17. Wang, P., Xiao, X. and Chou, K.-C. (2011) NR-2L: A Two-Level Predictor for Identifying Nuclear Receptor Subfamilies Based on Sequence-Derived Features. *PLoS One*, **6**, e23505. https://doi.org/10.1371/journal.pone.0023505
18. Cai, C.Z., Han, L.Y., Ji, Z.L., Chen, X. and Chen, Y.Z. (2003) SVM-Prot: Web-Based Support Vector Machine Software for Functional Classification of a Protein from Its Primary Sequence. *Nucleic Acids Research*, **31**,

3692-3697. https://doi.org/10.1093/nar/gkg600

19. Wei, L., Xing, P., Su, R., Shi, G., Ma, Z.S. and Zou, Q. (2017) CPPred-RF: A Sequence-Based Predictor for Identifying Cell-Penetrating Peptides and Their Uptake Efficiency. *Journal of Proteome Research*, **16**, 2044-2053. https://doi.org/10.1021/acs.jproteome.7b00019