J. Biomedical Science and Engineering, 2014, 7, 927-940
Published Online September 2014 in SciRes. http://www.scirp.org/journal/jbise
http://dx.doi.org/10.4236/jbise.2014.711090

Scientific Research

Predicting Beta-Turns and Beta-Turn Types Using a Novel Over-Sampling Approach

Lan Anh T. Nguyen[1,2], Xuan Tho Dang[1], Tu Kien T. Le[1], Thammakorn Saethang[1], Vu Anh Tran[1], Duc Luu Ngo[1], Sergey Gavrilov[1], Ngoc Giang Nguyen[1], Mamoru Kubo[3], Yoichi Yamada[3], Kenji Satou[3]

[1]Graduate School of Natural Science and Technology, Kanazawa University, Kanazawa, Japan
[2]Department of Computer Science, Hue University of Education, Hue, Vietnam
[3]Institute of Science and Engineering, Kanazawa University, Kanazawa, Japan
Email: lananh257@gmail.com, thodx@hnue.edu.vn, kienltt@hnue.edu.vn, thammakorn.kmutt@gmail.com, tvatva2002@gmail.com, ndluu@blu.edu.vn, gavriloff.sv@gmail.com, giangnn.bkace@gmail.com, mkubo@t.kanazawa-u.ac.jp, youichi@t.kanazawa-u.ac.jp, ken@t.kanazawa-u.ac.jp

Received 9 July 2014; revised 26 August 2014; accepted 10 September 2014

Abstract

β-turn is one of the most important reverse turns because of its role in protein folding. Many computational methods have been studied for predicting β-turns and β-turn types. However, due to the imbalanced dataset, the performance is still inadequate. In this study, we proposed a novel over-sampling technique FOST to deal with the class-imbalance problem. Experimental results on three standard benchmark datasets showed that our method is comparable with state-of-the-art methods. In addition, we applied our algorithm to five benchmark datasets from UCI Machine Learning Repository and achieved significant improvement in G-mean and Sensitivity. It means that our method is also effective for various imbalanced data other than β-turns and β-turn types.

Keywords

Beta-Turns, Beta-Turn Types, Class-Imbalance, Over-Sampling

1. Introduction

Secondary structure that includes regular and irregular patterns is important in protein folding study because it can be a building block of three-dimensional structures. The regular structures, which are sequences of residues with repeating φ and ψ values, are classified in α-helix and β-strand. While this group is well defined, the irregular structures that cover 50% of remaining protein residues are classified as coils. In fact, coils can be tight

How to cite this paper: Nguyen, L.A.T., Dang, X.T., Le, T.K.T., Saethang, T., Tran, V.A., Ngo, D.L., Gavrilov, S., Nguyen, N.G., Kubo, M., Yamada, Y. and Satou, K. (2014) Predicting Beta-Turns and Beta-Turn Types Using a Novel Over-Sampling Approach. *J. Biomedical Science and Engineering*, **7**, 927-940. http://dx.doi.org/10.4236/jbise.2014.711090

turns, bulges, or random coils. Among them, tight turn is the most important one from the viewpoint of protein structure as well as function [1]. Tight turns are categorized as δ-, γ-, β-, α-, and π-turns according to the number of consecutive residues in the turn.

β-turn is one of the most common tight turns. It is composed of four consecutive residues that are not in an α-helix and the distance between the first and the fourth C_α is less than 7 Å [1]. β-turns play an important role in the conformation as well as the function of protein, and make up around 25% of the protein residues. β-turns are the essential part of β-hairpins, provide the directional change of the polypeptide [2], and take part in the molecular recognition processes [3]. In addition, the formation of β-turn is a vital step in protein folding [4]. Therefore, the knowledge of β-turn is necessary in the three-dimensional structure prediction of a given primary protein sequence.

In addition, β-turns are further classified into some types according to the difference in three-dimensional structures. Based on the dihedral angles of the second and third residues in a β-turn, Hutchinson and Thornton proposed nine types of β-turn: I, I', II, II', IV, VIa1, VIa2, VIb, and VIII [5]. Because the types VIa1, VIa2, and VIb are rare, they are often combined into one type and named VI [1].

Prediction of β-turn by machine learning techniques have been studied actively, for instance, by Artificial Neural Network (ANN) [6]-[8], Support Vector Machines (SVMs) [3] [9]-[14], logistic regression [14] [15], and so on. In the realm of β-turn types prediction, most methods are based on ANN [6] [16], probabilities with multiple sequence alignments as COUDES [17], or SVMs [9] [18] [19]. However, the quality of β-turns and β-turn types prediction is still inadequate. One of the reasons is the small proportion of β-turn-residues in protein sequence. This is so-called the class-imbalance problem and often appears in Bioinformatics. The class-imbalance problem, in the serious case, causes the undesirable result that only majority class is correctly predicted.

Among many methods to handle the class-imbalance problem, resampling-based techniques including under-sampling and over-sampling methods are said to improve the classification performance significantly [20]. In this study, we propose a novel over-sampling method to deal with the class-imbalance problem in predicting β-turns and their types. Our algorithm generates the synthetic samples flexibly, for samples with minority samples as nearest neighbors as well as samples surrounded by majority samples. In addition, the new samples are informative and synthesized in a safe area. We present the experimental results on three standard benchmark datasets compared with state-of-the-art β-turns and β-turn types prediction methods. We also evaluate the performance of the novel over-sampling algorithm on the five other datasets from UCI Machine Learning Repository.

2. Materials and Methods

2.1. Datasets

We chose a benchmark dataset BT426 for the performance evaluation of our β-turn prediction method. It has been used in many researches [3] [6]-[10] [12] [13] [15] as the standard dataset for the comparison. In addition, two more other datasets, BT547and BT823, that were constructed for training and testing COUDES [17], were also used in our study. These datasets contain 426,547 and 823 protein sequences, respectively. All these protein chains have at least one β-turn and the similarity of each pair of chains is less than 25%. The observed turns and turn types in protein sequences were assigned by PROMOTIF program [21]. **Table 1** presents the ratio of residues belonging to β-turn or β-turn type i to the non-β-turn or non-β-turn type i (i = I, I', II, II', IV, VI, VIII) in these datasets.

2.2. Features

In this work, PSSMs (Position Specific Scoring Matrices), predicted shape strings, and predicted protein blocks

Table 1. The ratio of β-turn/β-turn type i (i = I, I', II, II', IV, VI, VIII) residues to the rest of protein residues in three standard benchmark datasets.

Dataset	Turn/non-turn	Type I	Type I'	Type II	Type II'	Type IV	Type VI	Type VIII
BT426	1:3.11	1:9.46	1:76.16	1:25.97	1:142.61	1:9.54	1:183.90	1:35.38
BT547	1:2.92	1:9.06	1:68.75	1:23.65	1:131.60	1:9.15	1:160.00	1:32.83
BT823	1:3.00	1:9.12	1:67.08	1:24.23	1:128.21	1:9.25	1:153.20	1:35.92

were used as the input features to predict β-turns and their types.

2.2.1. PSSMs

PSSMs were generated by using PSI-BLAST [22] against National Center for Biotechnology Information (NCBI) non-redundant sequence database with default parameters. PSSM is a matrix of N rows corresponding to the length of the protein sequence and 20 columns corresponding to 20 kinds of standard amino acids.

2.2.2. Predicted Shape Strings

Each residue in a protein sequence can be categorized into one of eight groups that are symbolized by eight symbols (S, R, U, V, K, A, T, and G) according to the phi-psi torsion angles. A sequence of these symbols makes up a shape string of a corresponding protein. The authors in [14] [23] used predicted shape strings to enhance the beta-turn prediction result.

In this work, DSP program [24] was used to predict the shape strings. Besides the eight states above, N is used to represent the residue with undefined phi-psi angles. Each state was encoded as a vector of nine features (1,0,0,0,0,0,0,0,0) for S, (0,1,0,0,0,0,0,0,0) for R, and so on.

2.2.3. Predicted Protein Blocks

Though predicted secondary structures of protein were effective in predicting β-turns and their types [7] [9] [12] [14] [23], the way of classifying a secondary structure of protein into three states of backbone conformation as α-helix, β-sheet, and coil leads to the circumstance that 50% total number residues are assigned as coils while they are believed to belong to a large set of distinct local structures [25] [26]. Therefore, the structural alphabets (SAs), that are sets of specific prototypes approximating the local protein structure, were developed to overcome this drawback [25].

Protein blocks, that allow a good approximation of local protein 3D structures [27] [28], have been utilized in many applications [26] [29]. SAs for protein blocks are sixteen pentapeptide motifs with labels A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, and P. Each of these prototypes represents a vector of eight average phi-psi angles.

In this research, predicted protein blocks were obtained from the website of PB-kPRED (http://www.bo-protscience.fr/pentapept/?page_id=9). Sixteen characters from A to P symbolize sixteen corresponding blocks and *X* for the other state. For each residue *i* in a protein chain, the corresponding predicted protein block was represented by a vector of seventeen features $\left(x_i^1, x_i^2, \cdots x_i^{17}\right)$, where x_i^j was the probability of residue *i* as state *j*.

The feature vector of each query residue was generated by using a sliding window of size nine amino acids. Thus, one input vector contained 414 attributes, where each PSSM value *x* was scaled to the range $[0,1]$ by the logistic function:

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

and the other values were normalized.

2.3. Methods

2.3.1. Resampling Techniques

Resampling techniques are said to effectively improve classification accuracy of the imbalanced datasets. While under-sampling methods decrease the number of majority samples, over-sampling methods enlarge the number of minority samples to rebalance the imbalanced dataset. However, the removal of samples may cause the significant information loss for the majority class. This is the main drawback of under-sampling methods. In contrast, over-sampling methods synthesize new minority samples in various ways. The most naïve method is random over-sampling that randomly chooses and replicates some minority samples. This method is simple, but often results in over-fitting. Another well-known over-sampling method is SMOTE [30], which generates the new samples by using the information of each minority sample and its randomly chosen minority nearest neighbor. The synthetic minority samples are located between these two minority samples considered. Therefore, SMOTE can improve the quality of synthetic samples; however, it may lead to the overlapping in classes. This problem becomes more serious when the original imbalanced dataset contains many isolated minority samples, which are the samples surrounded by the majority samples. To alleviate both problems of over lapping and over-fitting, we

propose a novel over-sampling method named Flexible Over-Sampling Technique (FOST).

2.3.2. Flexible Over-Sampling Technique

The main idea of FOST is to improve the density of each minority sample flexibly depending on the number of its nearest neighbors which belong to minority class. First of all, as shown in the pseudo-code below, FOST finds *k* nearest neighbors for each minority sample *x* (line 3). Note that the *k* nearest neighbors here can be minority samples, majority samples, or synthetic samples generated by the function **Self_sample_generation** (line 6) or **Sample_generation** (line 12). FOST synthesizes the new samples for *x* as follows:

- If x' is the nearest neighbor of *x* and x' and belongs to the majority class (line 5), FOST generates *d* synthetic samples opposite to x' so that the distances from these samples to *x* are less than the distance between *x* and x' (lines 16 - 22).
- If there are *m* (*m* < *k*) minority nearest neighbors of *x* and the distances from these samples to *x* are less than the distance from x' to *x* (line 8), FOST synthesizes *d* new samples as follows: 1) computes the sample *y* that is the centroid of *m* minority nearest neighbors and *x* (line 9); 2) generates a sample between *y* and one randomly chosen sample among its (*m* + 1) nearest neighbors (lines 23 - 27).
- If all *k* nearest neighbors of *x* belong to the minority class, FOST does not generate any synthetic sample.

The pseudo-code for FOST algorithm is as follows:

2.3.3. FOST Algorithm

Input: Minority dataset *M*; Majority dataset *N*; ratio of generation *d*; threshold *k*;
Output: set of synthetic samples *S*;
Begin
1) $S = \phi$;
2) For each $x \in M$;
3) Find *k* nearest neighbors of *x* in $M \cup N \cup S$;
4) If exists a majority nearest neighbor x' among these *k* nearest neighbors of *x*;
5) If x' is the most nearest neighbor of *x*;
6) **Self_samples_generation** (x, x', d);
7) else;
8) $PN = \{t^1, t^1, \cdots t^m\}$: *m* minority samples whose distances to *x* are smaller than the distance between *x* and x', *m* < *k*;
9) $y = (y_1, y_1, \cdots y_n)$: the new minority sample where $y_i = \frac{1}{m+1}\left(\sum_j t_i^j + x_i\right), j = 1 \cdots m$;
10) For *l* = 1:*d*;
11) t^j: is randomly chosen from $\text{PN} \cup \{x\}$;
12) **Samples_generation** (y, t^j);
13) end_for;
14) Update *S*;
15) End_for;
End
Function Self_samples_generation (x, x', d);
Input: samples $x = (x_1, x_2, \cdots x_n, x_{CL})$; $x' = (x'_1, x'_2, \cdots x'_n, x'_{CL})$; number of new samples *d*;
Output: set of synthetic samples *new_spls_arr* of *x*;
Begin
16) For *i* = 1:*d*;
17) $new\_sample_{CL} = x_{CL}$;
18) For *j* = 1:*n*;
19) $new\_sample_j = x_j - \varepsilon \times (x'_j - x_j), \varepsilon \in (0,1)$;
20) end_for;
21) push (*new_spls_arr*, *new_sample*);
22) end_for;
End

Function Samples_generation (x, x');
Input: samples $x = (x_1, x_2, \cdots x_n, x_{CL})$; $x' = (x'_1, x'_2, \cdots x'_n, x'_{CL})$;
Output: set of synthetic samples *new_spls_arr* of *x*;
Begin
23) $new\_sample_{CL} = x_{CL}$;
24) for j = 1:n;
25) $new\_sample_j = x_j + \varepsilon \times (x'_j - x_j), \varepsilon \in (0,1)$;
26) end_for;
27) push (*new_spls_arr*, *new_sample*);
End

2.3.4. Performance Evaluation of the Method

Since the ratio of β-turn to non-β-turn samples is around 1:3, the datasets are imbalanced. Support Vector Machine (SVM) was used as the basis classifier in this study since it is said to be better than other standard classifiers in dealing with imbalanced dataset. Specifically, ksvm function in kernlab package for *R* software [31] with Gaussian RBF kernel was employed.

We conducted seven-fold cross validation to evaluate the performance of our method. Each dataset was divided into seven parts that contained the same number of positive samples. Then, the feature selection based on information gain ratio [32] was applied to reduce the redundant features and achieve the highest MCC. After that, FOST was used to relax the imbalance ratio of the datasets. We set the threshold k = 10 for every case. The ratio of over-sampling d was chosen via grid search in each case.

To predict β-turn types, we created the same architecture as the prediction of β-turns, except the goal of the prediction was the β-turn type i (i = I, I', II, II', IV). It means that the non-β-turn residues and the residues belonging to β-turn type j, $j \neq i$, were the negative samples. In the cases of type VI and VIII, due to the high imbalance ratio, we random-under-sampled to relax the imbalance ratio before applying feature selection and FOST.

Since a β-turn contains at least four consecutive residues, the output needed to be filtered by applying the following rules in order [6]:

1) Change isolated predicted non-turn to turn: tnt → ttt.
2) Change isolated predicted turn to non-turn: ntn → nnn.
3) Change the two non-turn neighbors of two successive turns to turns: nttn → tttt.
4) Change the two non-turn neighbors of three successive turns to turns: ntttn → ttttt.

These rules ensure that the length of every final predicted turn is at least four residues.

Figure 1 demonstrates the overall architecture of our prediction method.

2.3.5. Performance Metrics

As MCC, Q_{total}, $Q_{observed}$, $Q_{predicted}$ are often used to measure the quality of β-turn prediction methods [17], they are used to evaluate the performance of our method and are defined as below:

$$\text{Matthews correlation coefficient}(\text{MCC}) = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FN)\times(TP+FP)\times(TN+FP)\times(TN+FN)}} \quad (2)$$

$$Q_{\text{total}} = \text{Accuracy} = (TP+TN)/(TP+FN+TN+FP) \quad (3)$$

$$Q_{\text{observed}} = \text{Sensitivity} = TP/(TP+FN) \quad (4)$$

$$Q_{\text{predicted}} = TP/(TP+FP) \quad (5)$$

where *TP*, *TN*, *FP*, *FN* are the number of true positive, true negative, false positive and false negative samples, respectively.

MCC, which lies in $[-1,1]$ is used to evaluate the correlation of the predicted and the observed class labels. Three values of −1, 0, 1 correspond to the worst, the random and the best predictor, respectively. It is the most robust measure for β-turn prediction [9].

In addition, the threshold-independent measures ROC (Receiver Operating Characteristics) and AUC (Area Under the Curve), which are often used in bioinformatics [33], are adopted.

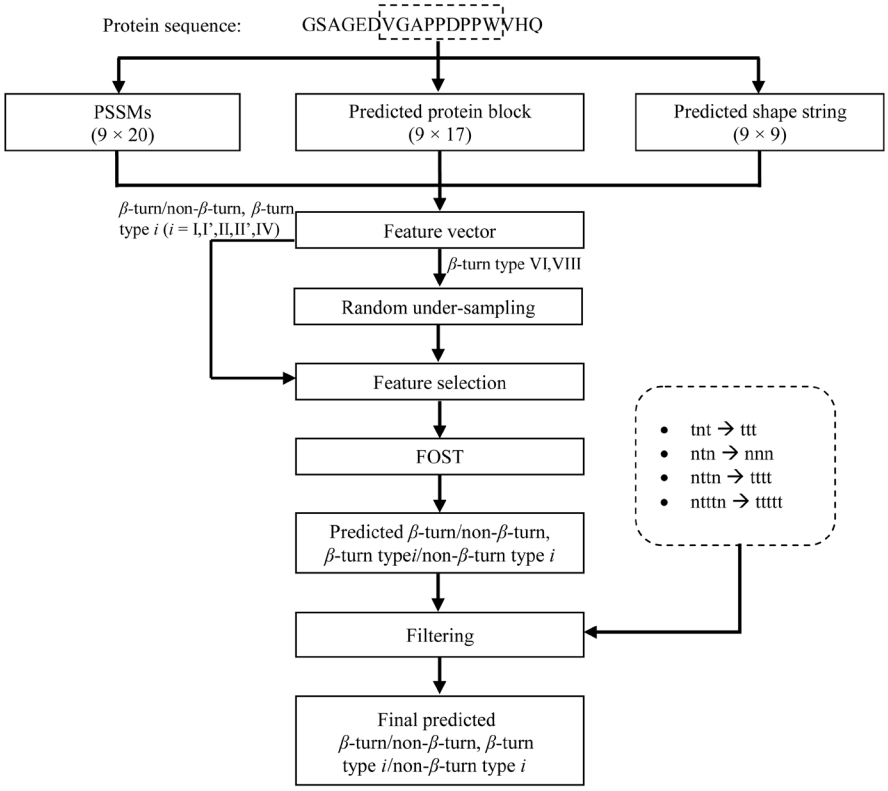

Figure 1. Scheme of our β-turns and β-turn types prediction method.

3. Results and Discussions

3.1. Prediction of Turn/Non-Turn

The proper choice of sliding window size for extracting the feature vectors affects the performance of prediction. Shepherd [6] showed that window of seven or nine residues was optimal for β-turn prediction. In the experiments, we tested various sliding window sizes and selected the size of nine residues since it returns not only the highest MCC but also the highest Q_{total}, Q_{observed}, and $Q_{\text{predicted}}$.

We also performed experiments to evaluate the impact of evolutionary information PSSMs, predicted protein blocks, and predicted shape string combinations. **Table 2** presents the effect of these feature groups on the BT426 dataset in predicting β-turn. The results show that using all three groups of features achieved the highest MCC, Q_{total}, Q_{observed}, $Q_{\text{predicted}}$, and AUC in comparison to using two of three of them. The highest performance results of the existing method that used PSSMs and predicted secondary structure as input features are 82.87%, 70.66%, 64.83%, 0.56, and 0.886 on Q_{total}, Q_{observed}, $Q_{\text{predicted}}$, MCC, and AUC, respectively [14]. In the case of using PSSMs and predicted protein blocks as input features, we achieved higher Q_{total}, $Q_{\text{predicted}}$, MCC, and AUC (1.85%, 5.01%, 0.03, and 0.007, respectively). It shows that predicted protein blocks are useful in identifying β-turns.

Figure 2 presents the ROC curves for predicting β-turn using the different combinations of feature groups on the BT426 dataset.

Our method outperformed the other competing methods with MCC of 0.66 except Tang *et al.* and H-SVM-LR. In comparison to Tang *et al.*, though MCC of the both methods was 0.66, we attained higher Q_{total} (87.48% vs. 87.2%) and $Q_{\text{predicted}}$ (75.26% vs. 73.8%). H-SVM-LR achieved higher MCC than us (0.01), but lower Q_{total} and $Q_{\text{predicted}}$ (87.37% vs. 87.48% and 74.99% vs. 75.26%, respectively). Note that while we applied the filtering to make the predicted beta-turn more realistic, H-SVM-LR did not. **Table 3** shows the results of all methods in detail.

Table 2 and **Table 3** show that the use of feature selection for eliminating redundant features and FOST to relax the class-imbalance, not only increase Q_{total} (0.9%), $Q_{\text{predicted}}$ (3.81%) but also MCC (0.02).

Table 2. The comparative results of different feature groups using ksvm on the BT426 dataset.

Feature group	Q_{total} (%)	$Q_{observed}$ (%)	$Q_{predicted}$ (%)	MCC	AUC
PSSMs + Predicted protein blocks	84.72	68.12	69.84	0.59	0.893
PSSMs + Predicted shape strings	85.74	71.20	70.44	0.61	0.900
PSSMs + Predicted protein blocks + Predicted shape strings	86.58	74.55	71.45	0.64	0.915

Table 3. Comparison of competing methods on the BT426 dataset.

Method	Q_{total} (%)	$Q_{observed}$ (%)	$Q_{predicted}$ (%)	MCC
Our method	87.48	72.24	75.26	0.66
H-SVM-LR [14]	87.37	75.20	74.99	0.67
Tang *et al.* [23]	87.2	75.9	73.8	0.66
KLR [15]	80.4	65.25	58.98	0.50
NetTurnP [8]	78.2	75.6	54.4	0.50
DEBT [9]	79.2	70.1	54.8	0.48
BTNpred [13]	80.9	55.6	62.7	0.47
SVM [12]	79.8	68.9	55.6	0.47
BTSVM [10]	78.7	62.0	56.0	0.45
BetaTPred [7]	75.5	72.3	49.8	0.43
BTPRED [6]	74.9	48.0	55.3	0.35

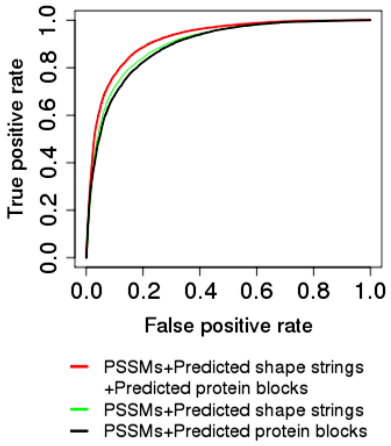

Figure 2. ROC curves for different feature groups on the BT426 dataset.

Figure 3 displays the ROC curve of our method, with the AUC was 0.921.

Besides the BT426 dataset, we performed the experiments on two more additional datasets, BT547 and BT823. **Table 4** presents the results of our method on the datasets BT547 and BT823 with MCCs of 0.66 and 0.67, respectively. The ROC curves of these two datasets are shown in **Figure 4**.

3.2. Prediction of β-Turn Types

The performance of our method in predicting β-turn types on the three datasets BT426, BT547, and BT823 is shown in **Table 5**. All the AUC values are higher than 0.7, and most of them are higher than 0.85. It proves our method is acceptable in predicting β-turn types [15].

Table 4. Turn/non-turn prediction results of our method on the BT547 and BT823 datasets.

Dataset	Q_{total} (%)	$Q_{observed}$ (%)	$Q_{predicted}$ (%)	MCC	AUC
BT547	87.18	74.38	75.12	0.66	0.921
BT823	87.76	74.14	76.22	0.67	0.925

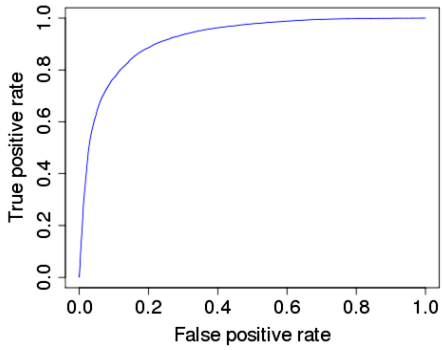

Figure 3. ROC curve of our method on the BT426 dataset.

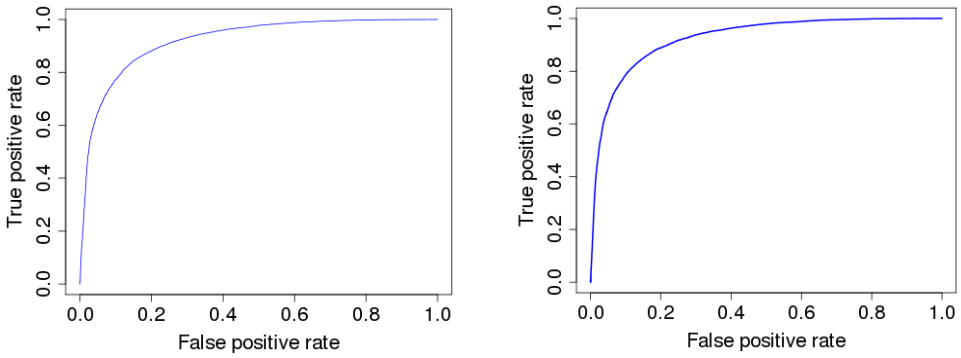

Figure 4. ROC curves of our method on the BT547 (a) and BT823 (b) datasets.

Table 6 presents the MCCs of the competing methods. Only our method could predict type VI. While DEBT could not predict types I' and II', our method achieved the highest MCC in comparison with other methods on all three datasets (0.75 and 0.64 on BT426; 0.78 and 0.66 on BT547; 0.80 and 0.66 on BT823 for type I' and II', respectively). Our method also achieved the highest MCC in predicting type II and VIII (0.75 and 0.30 on BT426; 0.77 and 0.35 on BT547; 0.77 and 0.33 on BT823). For type I, our method resulted in lower MCC in comparison to Shi *et al.* on BT426 (0.61 vs. 0.71), and equally on BT823 (0.64), but higher on BT547 (0.63 vs. 0.53). In the case of type IV, Shi *et al.* was the winner on BT426 (0.46), but our method achieved the best MCCs on BT547 (0.40) and BT823 (0.40). ROC curves of our β-turn types prediction are shown in **Figure 5**.

3.3. Datasets from UCI Machine Learning Repository

In addition to three standard benchmark datasets above, we evaluated the performance of our novel over-sampling algorithm FOST on the five datasets which were obtained from UCI Machine Learning Repository [34]: Haberman's Survival, Pima Indian Diabetes, Glass Identification, Landsat Satellite, and Yeast. The details of these datasets are described in **Table 7**.

The experiments were implemented to compare our method with the control method (*i.e.* no over-sampling) and SMOTE, using ksvm as the classifier with Gaussian RBF kernel and default parameters. We conducted the 10 independent times of 10-fold cross-validation on every dataset and averaged to get the performance of the methods. The optimal number of synthetic samples of each dataset for FOST algorithm was decided by grid

Table 5. Beta-turn types prediction results of our method on the BT426, BT547, and BT823 datasets.

Dataset	β-turn Type	Q_{total} (%)	$Q_{observed}$ (%)	$Q_{predicted}$ (%)	AUC
BT426	Type I	93.45	62.25	66.77	0.933
	Type I'	99.28	84.83	67.76	0.985
	Type II	97.90	85.75	67.99	0.983
	Type II'	99.44	70.49	58.16	0.976
	Type IV	90.18	39.16	47.83	0.823
	Type VI	98.07	26.08	8.43	0.880
	Type VIII	90.18	64.85	16.76	0.892
BT547	Type I	93.54	64.29	68.89	0.935
	Type I'	99.36	80.44	76.05	0.989
	Type II	98.05	85.44	71.77	0.982
	Type II'	99.42	74.22	58.94	0.982
	Type IV	89.78	43.18	47.92	0.832
	Type VI	89.08	54.20	3.07	0.847
	Type VIII	92.28	63.85	22.10	0.916
BT823	Type I	93.65	66.25	68.46	0.940
	Type I'	99.36	85.88	74.65	0.991
	Type II	98.10	84.82	72.12	0.983
	Type II'	99.45	68.60	63.35	0.978
	Type IV	90.51	38.88	51.80	0.828
	Type VI	96.36	36.37	6.81	0.869
	Type VIII	89.90	74.37	17.63	0.916

Table 6. MCCs comparison between the competing methods in predicting β-turn types on the BT426, BT547, and BT823 datasets.

Dataset	Method	Type I	Type I'	Type II	Type II'	Type IV	Type VI	Type VIII
BT426	Our method	0.61	0.75	0.75	0.64	0.38	0.14	0.30
	Shi *et al.* [18]	0.71	0.51	0.68	0.42	0.46	-	0.25
	NetTurnP [8]	0.36	0.23	0.31	0.16	0.27	-	0.16
	DEBT [9]	0.36	-	0.29	-	0.27	-	0.14
	COUDES [17]	0.31	0.23	0.30	0.11	0.11	-	0.07
BT547	Our method	0.63	0.78	0.77	0.66	0.40	0.11	0.35
	Nakamura *et al.* [19]	0.40	-	0.31	-	0.38	-	0.26
	Shi *et al.* [18]	0.53	0.54	0.55	0.34	0.31	-	0.04
	DEBT [9]	0.38	-	0.33	-	0.27	-	0.14
BT823	Our method	0.64	0.80	0.77	0.66	0.40	0.15	0.33
	Nakamura *et al.* [19]	0.37	-	0.30	-	0.35	-	0.17
	Shi *et al.* [18]	0.64	0.42	0.63	0.36	0.32	-	0.13
	DEBT [9]	0.39	-	0.33	-	0.27	-	0.14

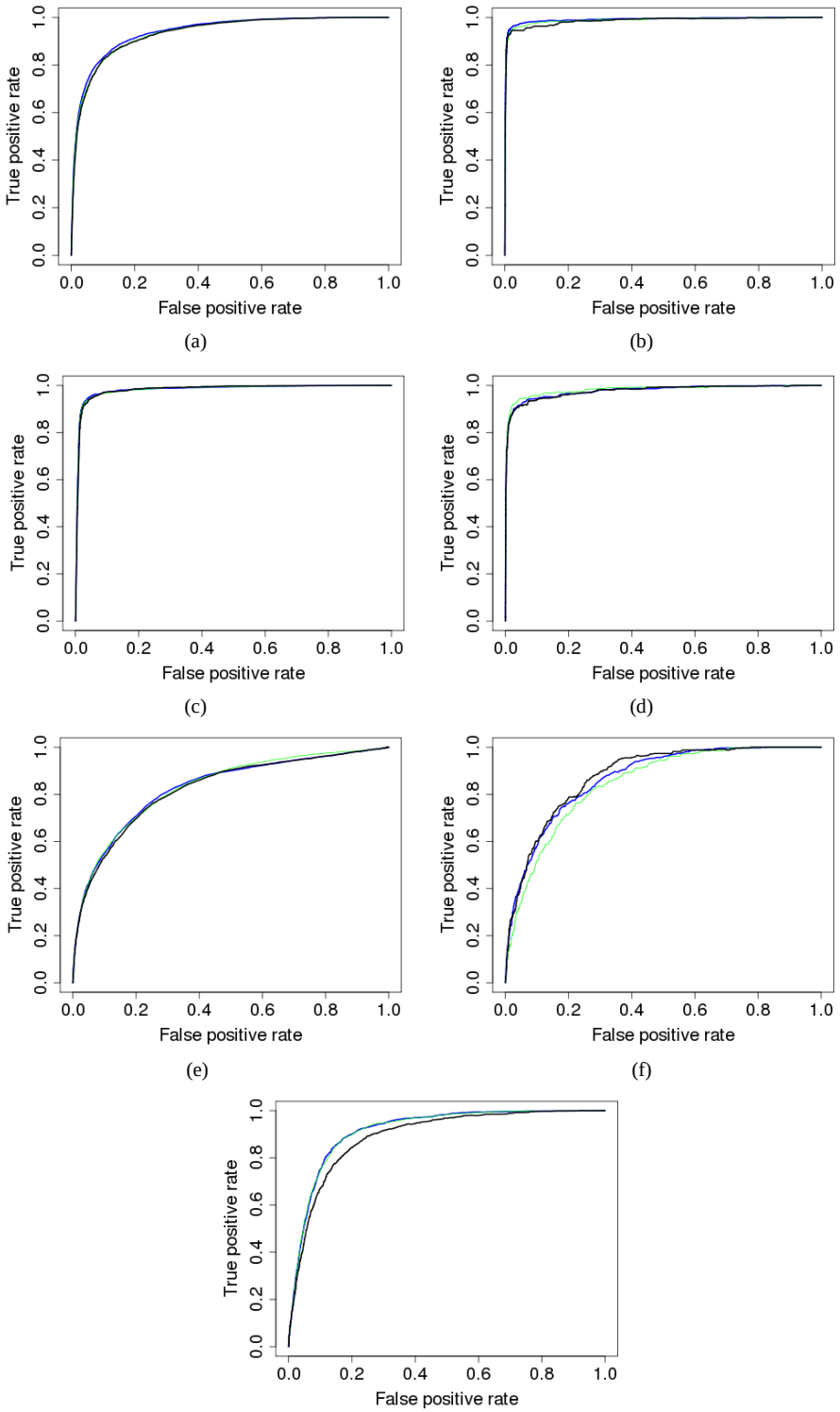

Figure 5. ROC curves of our method in predicting β-turn types on the three datasets BT426 (black), BT547 (green), and BT823 (blue). (a) type I; (b) type I'; (c) type II; (d) type II'; (e) type IV; (f) type VI; and (g) type VIII.

search, and then applied for SMOTE. **Table 8** presents the Accuracy, Sensitivity, Specificity, and G-mean of the

Table 7. The descriptions of the UCI datasets.

Dataset	Number of samples	Attributes	Minority class label	Imbalance ratio
Haberman's Survival	306	3	patient died within 5 years	1:2.78
Pima Indian Diabetes	768	8	tested positive for diabetes	1:1.87
Glass Identification	214	9	Headlamps	1:6.38
Landsat Satellite	6435	36	Damp grey soil	1:9.28
Yeast	1484	8	ME2	1:28.10

Table 8. The comparison of competing methods on the UCI datasets.

Dataset	Method	Accuracy (%)	Sensitivity (%)	Specificity (%)	G-mean (%)
Haberman's Survival	No Over-Sampling	73.63	19.01	93.29	42.00
	SMOTE	61.86	58.64	63.02	60.74
	FOST	60.00	74.20	54.89	63.80
Pima Indian Diabetes	No over-sampling	75.95	55.07	87.14	69.27
	SMOTE	73.87	72.69	74.50	73.58
	FOST	74.09	77.09	72.48	74.75
Glass Identification	No over-sampling	96.26	72.41	100.00	85.10
	SMOTE	94.86	76.55	97.73	86.49
	FOST	96.21	77.24	99.19	87.52
Landsat Satellite	No over-sampling	93.50	51.49	98.02	71.04
	SMOTE	93.78	67.51	96.61	80.67
	FOST	92.79	75.72	94.63	84.65
Yeast	No over-sampling	96.71	4.31	100.00	20.22
	SMOTE	95.91	29.61	98.27	47.20
	FOST	91.37	63.92	92.35	76.81

competing methods. Specificity and G-mean are defined as follows:

$$\text{Specificity} = TN/(TN + FN) \tag{6}$$

$$G\text{-mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}} \tag{7}$$

We also performed the two-sample t-test with equal variance to assess if the average G-means of different methods are significantly different. **Table 9** presents the p-values of these t-test comparisons between each pair of corresponding methods. All the p-values are smaller than 0.05, it means that our method achieves the better G-mean on all five benchmark datasets.

4. Conclusion

In this study, we presented a new method to identify β-turns and β-turn types in protein sequences. We showed that the use of predicted protein blocks as the input features well affected the prediction results. We also proposed a novel over-sampling algorithm FOST to relax the class-imbalance for the β-turn datasets effectively and improve the prediction performance. The combination of our new algorithm and the protein blocks features led to the significant improvement in prediction of β-turn types, especially, could predict type VI which is often

Table 9. The assessment by two-sample t-test with equal variance.

Dataset		No Over-Sampling	SMOTE
Haberman's Survival	No Over-Sampling	-	-
	SMOTE	6.39E−12	-
	FOST	1.37E−13	1.354E−03
Pima Indian Diabetes	No Over-Sampling	-	-
	SMOTE	1.736E−12	-
	FOST	1.901E−14	3.175E−05
Glass Identification	No Over-Sampling	-	-
	SMOTE	3.261E−06	-
	FOST	2.537E−07	1.195E−02
Landsat Satellite	No Over-Sampling	-	-
	SMOTE	1.243E−08	-
	FOST	<2.2E−16	7.159E−04
Yeast	No Over-Sampling	-	-
	SMOTE	6.2E−03	-
	FOST	<2.2E−16	2.867E−03

ignored by the previous methods. The experimental results on the UCI Machine Learning Repository datasets showed that FOST achieved better G-mean and Sensitivity than the control method and SMOTE with p-values less than 0.05. It means that our method is also effective for various imbalanced data other than β-turns and β-turn types.

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). The authors also wish to thank Dr. Osamu Hirose, Graduate School of Natural Science and Technology, Kanazawa University, Japan for his valuable comments.

References

[1] Chou, K.C. (2000) Prediction of Tight Turns and Their Types in Proteins. *Analytical Biochemistry*, **286**, 1-16. http://dx.doi.org/10.1006/abio.2000.4757

[2] Marcelino, A.M.C. and Gierasch, L.M. (2008) Roles of Beta-Turns in Protein Folding: From Peptide Models to Protein Engineering. *Biopolymers*, **89**, 380-391. http://dx.doi.org/10.1002/bip.20960

[3] Guruprasad, K. and Rajkumar, S. (2000) Beta-and Gamma-Turns in Proteins Revisited: A New Set of Amino Acid Turn-Type Dependent Positional Preferences and Potentials. *Journal of Biosciences*, **25**, 143-156.

[4] Takano, K., Yamagata, Y. and Yutani, K. (2000) Role of Amino Acid Residues at Turns in The Conformational Stability and Folding of Human Lysozyme. *Biochemistry*, **39**, 8655-8665. http://dx.doi.org/10.1021/bi9928694

[5] Hutchinson, E.G. and Thornton, J.M. (1994) A Revised Set of Potentials for Beta-Turn Formation in Proteins. *Protein Science*, **3**, 2207-2216. http://dx.doi.org/10.1002/pro.5560031206

[6] Shepherd, A.J., Gorse, D. and Thornton, J.M. (1999) Prediction of the Location and Type of Beta-Turns in Proteins Using Neural Networks. *Protein Science*, **8**, 1045-1055. http://dx.doi.org/10.1110/ps.8.5.1045

[7] Kaur, H. and Raghava, G.P.S. (2003) Prediction of Beta-Turns in Proteins from Multiple Alignment Using Neural Network. *Protein Science*, **12**, 627-634. http://dx.doi.org/10.1110/ps.0228903

[8] Petersen, B., Lundegaard, C. and Petersen, T.N. (2010) NetTurnP—Neural Network Prediction of Beta-Turns by Use of Evolutionary Information and Predicted Protein Sequence Features. *PLoS ONE*, **5**, e15079. http://dx.doi.org/10.1371/journal.pone.0015079

[9] Kountouris, P. and Hirst, J.D. (2010) Predicting Beta-Turns and Their Types Using Predicted Backbone Dihedral Angles and Secondary Structures. *BMC Bioinformatics*, **11**, Article ID: 407. http://dx.doi.org/10.1186/1471-2105-11-407

[10] Pham, T.H., Satou, K. and Ho, T.B. (2003) Prediction and Analysis of Beta-Turns in Proteins by Support Vector Machine. *Genome Informatics*, **14**, 196-205.

[11] Zhang, Q., Yoon, S. and Welsh, W.J. (2005) Improved Method for Predicting β-Turn Using Support Vector Machine. *Bioinformatics*, **21**, 2370-2374. http://dx.doi.org/10.1093/bioinformatics/bti358

[12] Hu, X. and Li, Q. (2008) Using Support Vector Machine to Predict β- and γ-Turns in Proteins. *Journal of Computational Chemistry*, **29**, 1867-1875. http://dx.doi.org/10.1002/jcc.20929

[13] Zheng, C. and Kurgan, L. (2008) Prediction of β-Turns at Over 80% Accuracy Based on an Ensemble of Predicted Secondary Structures and Multiple Alignments. *BMC Bioinformatics*, **9**, 430. http://dx.doi.org/10.1186/1471-2105-9-430

[14] Elbashir, M., Wang, J., Wu, F.X. and Wang, L. (2013) Predicting β-Turns in Proteins Using Support Vector Machines with Fractional Polynomials. *Proteome Science*, **11**, S5. http://dx.doi.org/10.1186/1477-5956-11-S1-S5

[15] Elbashir, M.K., Wang, J., Wu, F. and Li, M. (2012) Sparse Kernel Logistic Regression for β-Turns Prediction. 2012 *IEEE* 6*th International Conference on Systems Biology* (*ISB*), Xi'an, 18-20 August 2012, 246-251.

[16] Kirschner, A. and Frishman, D. (2008) Prediction of β-Turns and β-Turn Types by a Novel Bidirectional Elman-Type Recurrent Neural Network with Multiple Output Layers (MOLEBRNN). *Gene*, **422**, 22-29. http://dx.doi.org/10.1016/j.gene.2008.06.008

[17] Fuchs, P.F.J. and Alix, A.J.P. (2005) High Accuracy Prediction of β-Turns and Their Types Using Propensities and Multiple Alignments. *Proteins*: *Structure, Function, and Bioinformatics*, **59**, 828-839. http://dx.doi.org/10.1002/prot.20461

[18] Shi, X., Hu, X., Li, S. and Liu, X. (2011) Prediction of β-Turn Types in Protein by Using Composite Vector. *Journal of Theoretical Biology*, **286**, 24-30. http://dx.doi.org/10.1016/j.jtbi.2011.07.001

[19] Nakamura, M., Kajiwara, Y., Otsuka, A. and Kimura, H. (2013) LVQ-SMOTE—Learning Vector Quantization Based Synthetic Minority Over-Sampling Technique for Biomedical Data. *BioData Mining*, **6**, 16.

[20] He, H. and Garcia, E.A. (2009) Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, **21**, 1263-1284. http://dx.doi.org/10.1109/TKDE.2008.239

[21] Hutchinson, E.G. and Thornton, J.M. (1996) PROMOTIF—A Program to Identify and Analyze Structural Motifs in Proteins. *Protein Science*, **5**, 212-220. http://dx.doi.org/10.1002/pro.5560050204

[22] Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D.J. (1997) Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs. *Nucleic Acids Research*, **25**, 3389-3402. http://dx.doi.org/10.1093/nar/25.17.3389

[23] Tang, Z., Li, T., Liu, R., Xiong, W., Sun, J., Zhu, Y. and Chen, G. (2011) Improving the Performance of β-Turn Prediction Using Predicted Shape Strings and a Two-Layer Support Vector Machine Model. *BMC Bioinformatics*, **12**, 283. http://dx.doi.org/10.1186/1471-2105-12-283

[24] Sun, J., Tang, S., Xiong, W., Cong, P. and Li, T. (2012) DSP: A Protein Shape String and Its Profile Prediction Server. *Nucleic Acids Research*, **40**, W298-W302. http://dx.doi.org/10.1093/nar/gks361

[25] Offmann, B., Tyagi, M. and de Brevern, A.G. (2007) Local Protein Structures. *Current Bioinformatics*, **2**, 165-202. http://dx.doi.org/10.2174/157489307781662105

[26] Joseph, A.P., Agarwal, G., Mahajan, S., Gelly, J.C., Swapna, L.S., Offmann, B., Cadet, F., Bornot, A., Tyagi, M., Valadié, H., Schneider, B., Etchebest, C., Srinivasan, N. and de Brevern, A.G. (2010) A Short Survey on Protein Blocks. *Biophysical Reviews*, **2**, 137-145. http://dx.doi.org/10.1007/s12551-010-0036-1

[27] De Brevern, A.G., Etchebest, C. and Hazout, S. (2000) Bayesian Probabilistic Approach for Predicting Backbone Structures in Terms of Protein Blocks. *Proteins*: *Structure, Function, and Bioinformatics*, **41**, 271-287. http://dx.doi.org/10.1002/1097-0134(20001115)41:3<271::AID-PROT10>3.0.CO;2-Z

[28] De Brevern, A.G. (2005) New Assessment of a Structural Alphabet. *In Silico Biology*, **5**, 283-289.

[29] Joseph, A.P., Srinivasan, N. and de Brevern, A.G. (2011) Improvement of Protein Structure Comparison Using a Structural Alphabet. *Biochimie*, **93**, 1434-1445. http://dx.doi.org/10.1016/j.biochi.2011.04.010

[30] Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. (2002) SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research*, **16**, 321-357.

[31] Karatzoglou, A., Wien, T.U., Smola, A., Hornik, K. and Wien, W. (2004) Kernlab—An S4 Package for Kernel Methods in R. *Journal of Statistical Software*, **11**, 1-20.

[32] Altidor, W., Khoshgoftaar, T.M. and Hulse, J.V. (2011) Robustness of Filter-Based Feature Ranking: A Case Study. *Proceedings of* 24*th Florida Artificial Intelligence Research Society Conference* (*FLAIRS*-24), Palm Beach, 18-20 May 2011, 453

[33] Sonego, P., Kocsor, A. and Pongor, S. (2008) ROC Analysis: Applications to the Classification of Biological Sequences and 3D Structures. *Briefings in Bioinformatics*, **9**, 198-209. http://dx.doi.org/10.1093/bib/bbm064

[34] Bache, K. and Lichman, M. (2013) UCI Machine Learning Repository. School of Information and Computer Sciences, University of California, Irvine.

Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

Other selected journals from SCIRP are listed as below. Submit your manuscript to us via either submit@scirp.org or Online Submission Portal.