Resampling Methods to Handle the Class-Imbalance Problems in Predicting Protein-Protein Interaction Site and Beta-Turn

NGUYEN THI LAN ANH

July, 2013

Dissertation

Resampling Methods to Handle the Class-Imbalance Problems in Predicting Protein-Protein Interaction Site and Beta-Turn

Graduate School of
Natural Science & Technology
Kanazawa University

Major subject:
Division of Electrical Engineering
and Computer Science

Course:
Intelligent Systems
and Information Mathematics

School registration No.: 1023112109
Name: NGUYEN THI LAN ANH
Chief advisor: Professor KENJI SATOU

Abstract

Proteins are the active functional biomolecules. They are responsible for many tasks in the cells, such as catalyzing the biochemical reactions, creating the cell walls, involving in the defending the body from foreign invaders, involving in the movement, and so on. Most proteins interact with the other proteins or molecules to perform their functions; only a small number of them can work alone.

Though many advances have been achieved in the field of genome biology and Bioinformatics, the functions of many protein sequences have not been determined until now. However, the functions of the unknown protein can be inferred from the functions of the known proteins that interact with it. In addition, functions of a protein directly depend on its three-dimensional structure. The understanding of protein is the understanding its sequence, structure and function. Therefore, studying of protein-protein interaction and protein structure is very important in bioinformatics and has been receiving a lot of interests.

The study of protein-protein interaction aims to localize where protein sequence can physically interact, and to predict which proteins interact with which others. The first problem is called protein-protein interaction sites prediction. Learning about this issue leads to the understanding how proteins recognize the other molecules.

Predicting β-turns and their types is one of the protein structure prediction problems, and also is one of the interesting and hard problems in bioinformatics in recent years. The purpose is to provide more information for fold recognition study. However, the performances of both β-turns prediction and protein-protein interaction sites prediction are still far from being perfect. One of the main reasons is the existence of class-imbalance problem in the datasets.

This thesis intends to enhance the performances of predicting (i) the protein-protein interaction site by relaxing the class imbalance problem utilizing our novel over-sampling method together with using predicted shape strings; and (ii) the β-turn and beta-turn's types applying PSSMs, predicted protein blocks and random under-sampling technique.

For the predicting protein-protein interaction sites problem, experimental results on the dataset that contains 2,829 interface residues and 24,616 non-interface residues showed a significant improvement of our method in comparison with the other state-of-the-art methods according to six evaluation measures.

We performed experiments on three standard benchmark datasets that contain 426, 547 and 823 protein sequences, respectively, to evaluate the performance of our method for predicting the β-turns and their types. The results showed the substantial improvement of our approach compared with the other strategies.

Acknowledgments

This thesis marks the end of my three years of studying in Japan. From the depth of my heart, I would like to take this opportunity to thank everyone, who has given me a lot of kind help all the time I have been here.

I am deeply grateful to my supervisor, Professor Kenji Satou, for everything he has given me from the first moment picking me up at the airport to date. I greatly appreciate him for his enthusiasm, his patience, and for always giving the valuable and insightful advices to me. I thank him for teaching me not only Bioinformatics but also Japanese and the knowledge about the world.

I am thankful to Doctor Osamu Hirose for giving insightful comments and suggestions. I would like to thank Professor Yoichi Yamada, Professor Mamoru Kubo for their support.

My deep thanks go to all the committee members, Professor Kenji Satou, Professor Haruhiko Kimura, Professor Tu Bao Ho, Associate Professor Yoichi Yamada, and Lecturer Hidetaka Nambo for reading my thesis and giving the constructive comments.

I am so proud and excited to be a part of Bioinformatics Laboratory, Kanazawa University. I would like to show my greatest appreciation to everyone for the collaboration. Especially thanks to Tho, Seathang, Vu Anh, Kien and Luu for the wonderful moments we had together.

I would like to offer my special thanks to all of my Japanese teachers and the staff of Kanazawa University for their enthusiasm; to my sincere Japanese friends for their kindness. My life here was absolute hard without their help.

My gratitude goes to all the members of Vietkindai for supporting and helping me.

I owe my deepest gratitude to my colleagues in the Department of Informatics, Hue University's College of Education, Hue University, especially to Mr. Nguyen Duc Nhuan, for their support. I never can finish my study without their help.

To my teacher, Doctor Hoang Thi Lan Giao, I am so grateful for her guidance, her care and her encouragement to me.

Thanks to my close friends for always being there for me.

Thanks to my little Vietnamese students. They are one of the reasons makes me keep trying.

Thanks to Freda. Though short, she made my days in Wakunami Shukusha be meaningful with friendship.

Many thanks go to my neighbors in Hinoki Apaato, Minh, Nguyen, Tu and Manh, who have treated me as a sister without any condition. Especially thanks for sharing food with me and listening to my talk whenever I need.

It can be longer than my thesis if I list all the people who have helped me to have today; but I always appreciate all.

And of course, my deepest appreciation goes to my Dad and Mom, my grandfather, my brother and sisters, to my little nieces. I never can thank enough for their sacrifice.

Thanks to beloved Vietnam for giving chances and welcome me back. Thanks to beautiful Japan for great experiences.

The last three years are the important part of my life and will go with me to the end; I will respect for both good and bad memories, and will keep in my heart forever.

Thank you so much!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we introduce some basic concepts related to our methods in the next chapters, such as protein structure levels, torsion angles, protein blocks, β-turn, and so on. After that, we briefly present some concepts and research problems of protein-protein interaction sites and β-turns and their types prediction. And then, class-imbalance problem, one of the difficulties in predicting protein-protein interaction site and β-turn is introduced. Dealing with these problems is our purpose. Finally, we show the contributions and organization of our thesis.

1.1 Introduction

1.1.1 Protein overview

Protein

Proteins are cellular large molecules that are constructed from chains of hundreds or thousands amino acids. Each chain is called a polypeptide. Each individual amino acid in this chain is called a residue. Two amino acids link together through the peptide bond.

There are 20 amino acids that most commonly occur in nature. All of them consist of the same part, but the side chain R, as in Figure 1.1

Figure 1.2 presents the way that two amino acids link together to form a dipeptide in a protein chain.

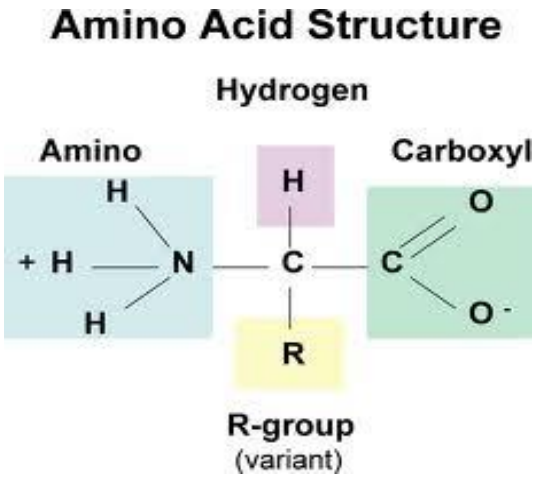


Figure 1.1 Basic structure of amino acid.

The different amino acids have the different side-chain R. (Figure adapted from http://sph.bu.edu/otlt/MPH-Modules/PH/PH709_A_Cellular_World/PH709_A_Cellular_World6.html)

Proteins play a very important role in the cells of living organisms. Each protein has a specific function, for example, enzymes catalyze the metabolic reactions; structural protein involves in creating the cell wall; regulatory proteins regulate the transcription of genes; transport proteins bring molecules traveling through the body; antibodies help to protect the body by binding to the specific foreign invaders such as bacteria or viruses, and so on.

Most proteins interact with the other molecules to perform their function. If the interactions between proteins in a cell disappear, the cell will be blind, deaf, paralytic and disintegrate.

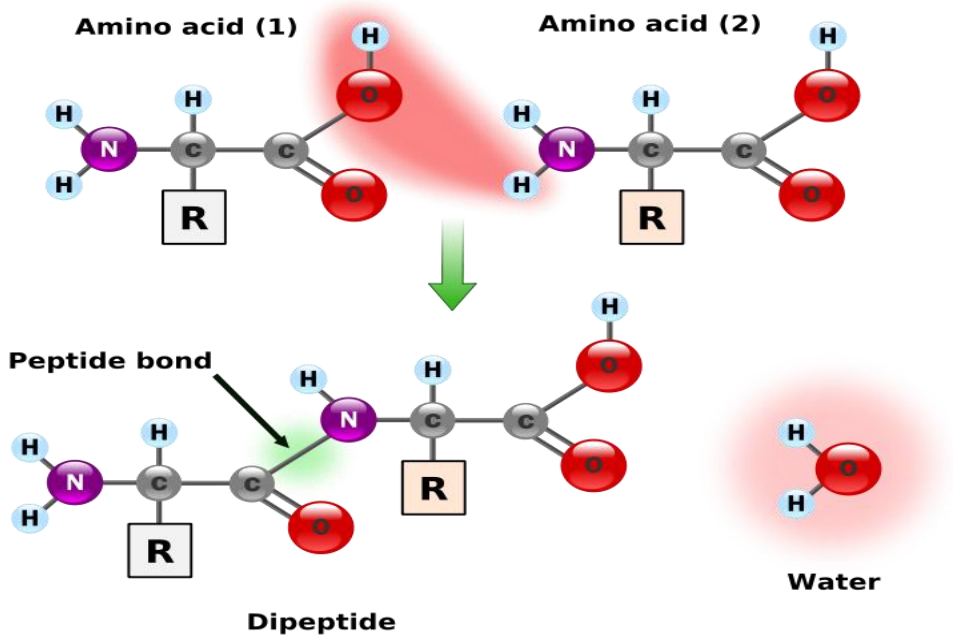


Figure 1.2 The condensation of two amino acids to form a dipeptide. (Figure adapted from ttp://en.wikibooks.org/wiki/An_Introduction_to_Molecular_Biology/Function_and_structure_of_Proteins)

Figure 1.3 presents an example of antibody Immunoglobulin G traveling in the blood and protecting the body by binding with the invaders.

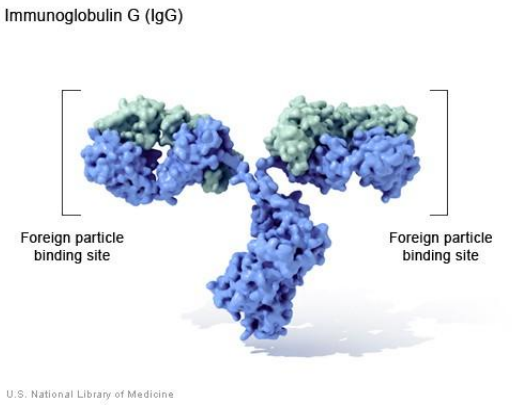


Figure 1.3 Antibody Immunoglobulin G recognizes foreign particles that might be harmful to defend the body.

(Figure downloaded from http://ghr.nlm.nih.gov/handbook/howgeneswork/protein)

Functions of proteins directly depend on their structure and shape. Protein structure can be presented as four levels (Figure 1.4):

- The primary structure is a linear amino acid sequence.
- Secondary structure refers to the local spatial arrangement of a polypeptide's backbone atoms without regard to the conformations of its side chains.
- Tertiary structure is the three-dimensional structure of an entire protein sequence.
- Some proteins contain more than one polypeptide chain. In this case, quaternary structure of a protein is the arrangement of the three-dimensional polypeptides.

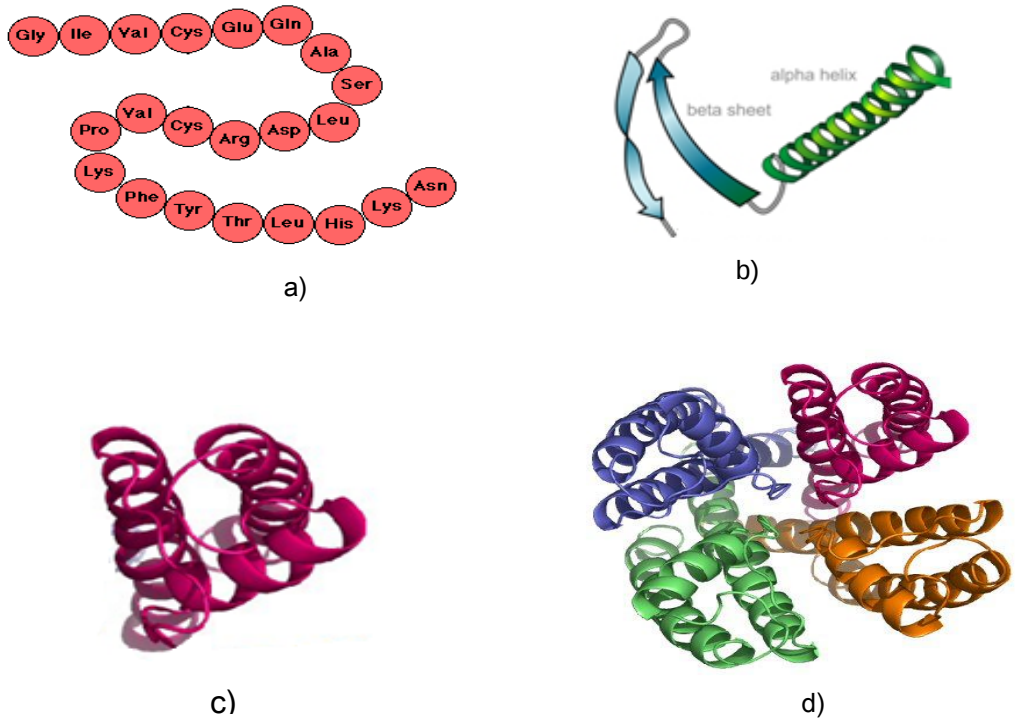


Figure 1.4 Four levels of protein structure.

a) Primary structure is a sequence of amino acids.

b) Secondary structure is the spatial arrangement of the specific regions.

c) Tertiary structure is the 3D structure of the whole polypeptide chain.

d) Quaternary structure, if exists, is the 3D structure of many polypeptide chains.

Torsion angles

The backbone (main chain) of a protein includes the atoms which participate in the peptide bonds. It can be displayed as a linked sequence of rigid planar peptide groups and described by the torsion angles (dihedral angles) ϕ and ψ. ϕ is the angle between two adjacent planes (CNCα) and (NCαC); and ψ is the angle between the planes (NCαC) and (CαCN) (Figure 1.5). These two angles are defined as 180° if the polypeptide sequence is fully extended conformation. Torsion angles are among the most important local structural parameters that control protein folding. If we know the values of these angles, we would be able to predict the corresponding protein 3D structure.

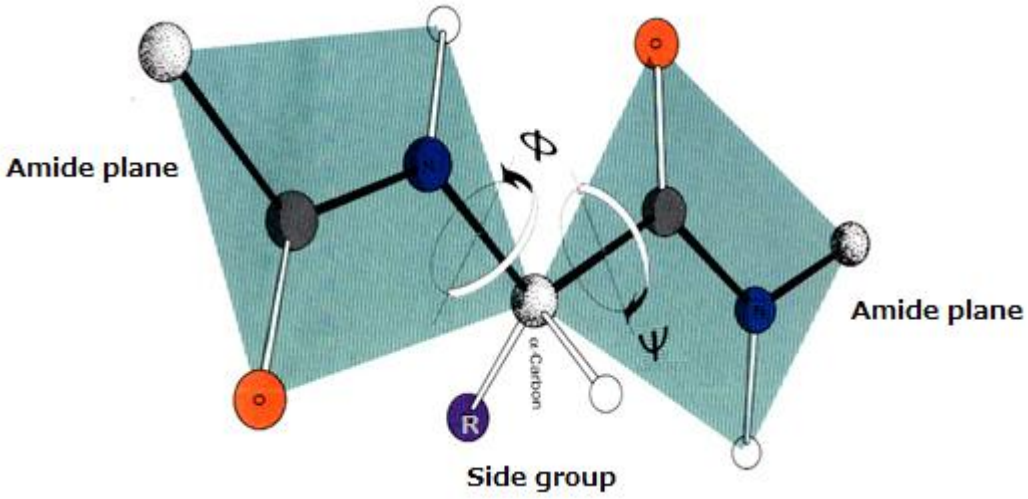


Figure 1.5 Torsion angles ϕ and ψ of the polypeptide backbone

Figure adapted from http://wiki.christophchamp.com/index.php/Ramachandran_plot

Protein blocks

Secondary structure of protein is very important for fold recognition. Secondary structures have been classically described into three states of backbone conformation as α-helix, β-sheet and coil. Around 50% of total number protein residues are assigned as coils. Meanwhile, these residues actually correspond to many distinct local protein structures. Therefore, a new view of three-dimensional protein structure that combines the small local fragments (or prototypes) has been developed. A structural alphabet (SA) is a complete set of these prototypes [1].

Because each residue relates to one of the fragments in a SA, a protein primary structure can be translated into a chain of prototypes in one dimension as the sequence of prototypes [2].

Many structural alphabets were developed, such as Building Blocks, Recurrent local structural motifs, Substructures, Structural Building Blocks, Oligons, Protein Blocks, LSP, Kappa-alpha, and so on. The more details can be found in [1].

Protein Blocks (PBs) [3] that allows a good approximation of local protein 3D structures [4] and has been applied to many applications at the present time [2, 5]. This SA is composed of sixteen local structure prototypes of five consecutive Cα, called Protein Blocks (PBs), labeled a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, respectively. Each of these prototypes represents a vector of eight average dihedral angles ϕ/ψ. Figure 1.6 displays these kinds of blocks.

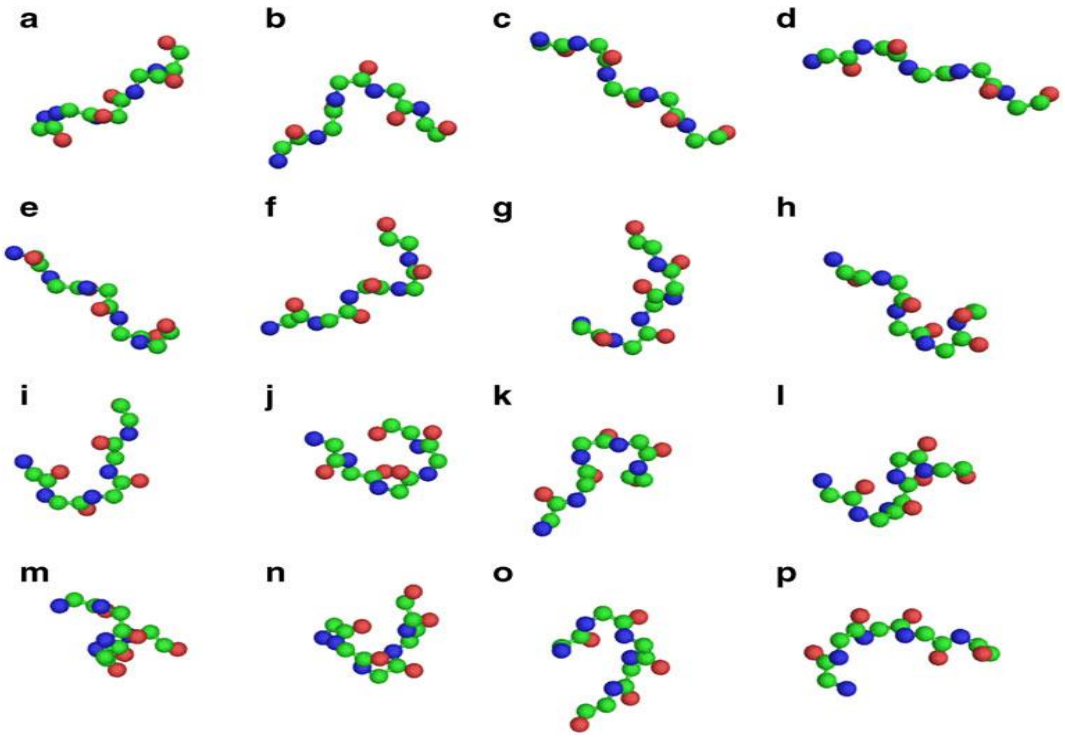


Figure 1.6 The protein blocks.

For each protein block, the N-cap extremity is shown on the left and the C-cap on the right. Each prototype is five residues in length and corresponds to eight dihedral angles (φ,ψ). The protein blocks m and d are mainly associated to the central region of α-helix and the central region of β-strand, respectively [2]

1.1.2 Protein-protein interaction sites prediction

Protein-protein interactions play a major role in maintaining normal cell functions and physiology [6]. Specifically, they are responsible for many important biological processes, such as metabolic control, DNA replication, protein synthesis, immunological recognition, and so forth. Thus, studying of protein-protein interaction is a vital task in bioinformatics. This realm contains two main goals, recognizing the interaction sites (or protein interfaces) where proteins physically contact, and predicting which pairs of proteins can interact. The knowledge of protein interfaces allows us to understand the way protein recognizes the other molecules and engineers new interactions. It is also very useful in identifying drug targets, designing drug-like peptides to prevent unwanted interactions [7, 8]. The demonstration of the interaction sites of two protein sequences is presented in Figure 1.7.

There are many experimental methods to identify the protein interaction sites and interface residues, such as X-ray Crystallography, Nuclear magnetic resonance [9] or Site-specific mutagenesis [10]. However, these approaches are expensive, time-consuming and problematic for transient complexes [11], while computational methods are more cost-effective.

Predicting protein-protein interaction sites by machine learning methods can be dealt as a classification problem that to predict whether an amino acid is an interface residue or not. The features that can distinguish interaction and non-interaction residues are used to describe protein site [11].

There are two main groups of methods for predicting protein-protein interaction sites, the methods using protein structure and the methods using protein sequence information [12].

The protein structure based methods represent each residue by information of its nearest neighbors in structure [13–15], thus they can utilize the informative features. However, the number of known-structure proteins to date is significantly smaller than the amount of protein sequences [16]. Therefore, it is necessary to develop the methods that can predict the interface residues from the amino acid sequence only, without knowing structural information. These methods generally generate the features for each residue from information of it and its neighbors in the sequence.

Some studies have attempted to develop the techniques for predicting interaction

sites from protein sequences. For example, Kini and Evans [17] relied on the most common appearance of proline in the flanking segments of interaction sites to propose the prediction method; Chen and Li [18] combined the hydrophobic and evolutionary information of amino acid to construct the prediction model; Chen and Jeong [16] extracted a wide range of features from protein sequences only and using Random Forests to create a prediction integrative model, and so forth.

However, it is not easy to apply sequence-based methods for interaction sites prediction due to the lack of understanding of biological properties that can provide vital information related to binding sites. Ofran and Rost [19, 20] proved that using better information would induce better prediction results. On the other hand, because the number of non-interacting residues is much more than the number of interacting residues, it often leads to the high value of false predicted negative.

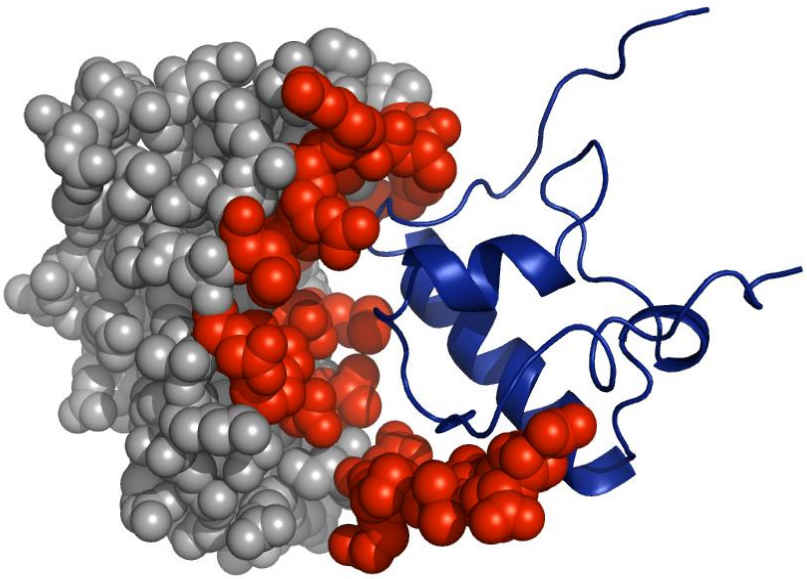


Figure 1.7 Illustration of protein-protein interaction interface residues of sequence 1FJG-F and ribosomal subunit S18.

Reds denote the interface residues.

(Figure adapted from http://www.insun.hit.edu.cn/~mhli/site_CRFs/fig/1FJG_F_right_1024.png**)**

1.1.3 β-turn prediction

There is a tight relationship between a protein sequence, structure, and its function. The understanding of structural basis for protein function can speed up the progress in systems biology that aims at identifying functional networks of proteins. For example, the rational drug design heavily relies on the structural knowledge of a protein [6].

Secondary structure, that includes regular and irregular patterns, is very important in protein folding study since it can provide the useful information to derive the possible three-dimensional structures. The regular structures, composed of sequences of residues with repeating φ and ψ values, classified in α-helix and β-strand. While this class is well defined, the other class, irregular structures, involves 50% of remaining protein residues are classified as coils. In fact, coil can be tight turn, bulge or random coil. Among of these structures, tight turn is the most important from the viewpoint of structure as well as function [21].

Tight turns are categorized into δ-turn, γ-turn, β-turn, α-turn and π-turn basing on the number of consecutive residues in the turn. Table 1.1 displays the kinds of tight turns.

β-turn is one of the most common tight turns. A β-turn is composed of four consecutive residues that are not in an α-helix and the distance between the first and the fourth Cα is less than 7Å [22] (Figure 1.8). β-turns play an important role in the conformation as well as the function of protein, and make up around 25% of the residue numbers. β-turns are the essential part of β-hairpins, provide the directional change of the polypeptide [23], and involve in the molecular recognition processes [24]. In addition, the formation of β-turn is a vital step in protein folding [25]. Therefore, the knowledge of β-turn is very necessary in the prediction of three-dimensional structure of a given primary protein sequence.

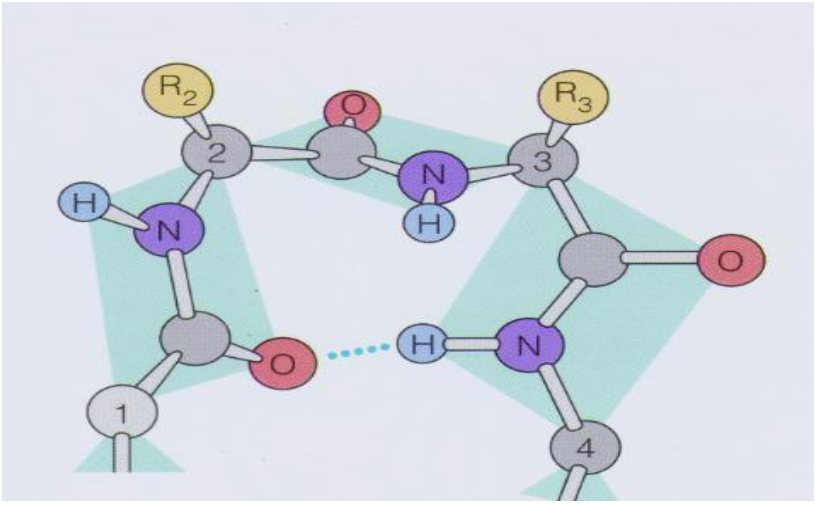


Figure 1.8 An example of beta-turn that contains four consecutive residues.

The C-α are numbered from 1 to 4. Dot line represents hydrogen bond

Table 1.1 Kinds of tight turns in protein

Type	No. of residues	H-bonding
δ-turn	2	NH(i)-CO(i+1)
γ-turn	3	CO(i)-NH(i+2)
β-turn	4	CO(i)-NH(i+3)
α-turn	5	CO(i)-NH(i+4)
π-turn	6	CO(i)-NH(i+5)

β-turns are categorized into nine types (I, I', II, II', IV, VIa1, VIa2, VIb and VIII) based on the dihedral angles of residues i+1 and i+2 in the turn [26]. The detailed values of these angles corresponding to each type are shown in Table 1.2. Because the turn types VIa1, VIa2 and VIb are rare, they are often combined into one type and named VI [21]. Figure 1.9 below displays the illustrative drawings of nine β-turn types.

The β-turn prediction methods can be divided into two main categories: statistical techniques and machine learning techniques. The former group includes the techniques such as Chou-Fasman's method [27], Thornton's methods [28, 29], Chou's method [30], the 1-4 and 2-3 correlation model [31] using the positional frequencies and β-turn residue conformation parameters; and the more recently method COUDES

[32], that used the propensities and multiple sequence alignments.

The latter group was reported to be effectively applied for β-turns prediction in recent years [33]. Belonging to this realm, Artificial Neural Network (ANN) was first used in [34], then frequently used by the other authors [22, 35, 36]. Support Vector Machines (SVMs) were also selected by many authors [24, 33, 37–41]. The most recent reported result is KLR, which used kernel logistic regression for prediction, with 0.5 on Matthews correlation coefficient (MCC) [42].

Most of the methods for the turn types prediction are based on ANN [35, 43, 44] or probabilities with multiple sequence alignments as COUDES [32]. More recently, Kountouris and Hirst [33] and X.Shi [45] used SVM in their methods and achieved the significant results. However, the quality of both β-turn location and turn types prediction is a challenge.

Table 1.2 Average values of dihedral angles of beta-turn types.

The third residue of turns type VIa1,VIa2, VIb must be a proline [21, 26]

Type	Dihedral angles (°)				$C^{\alpha}(i) - C^{\alpha}(i+3)$ distance (Å)
	φ_{i+1}	ψ_{i+1}	φ_{i+2}	ψ_{i+2}	
I	-60	-30	-90	0	4.6
I’	60	30	90	0	4.6
II	-60	120	80	0	4.6
II’	60	-120	-80	0	4.6
IV	-61	10	-53	17	7.2
VIa1	-60	120	-90	0	3.4
VIa2	-120	120	-60	0	3.7
VIb	-135	135	-175	160	6.0
VIII	-60	-30	-120	120	6.3

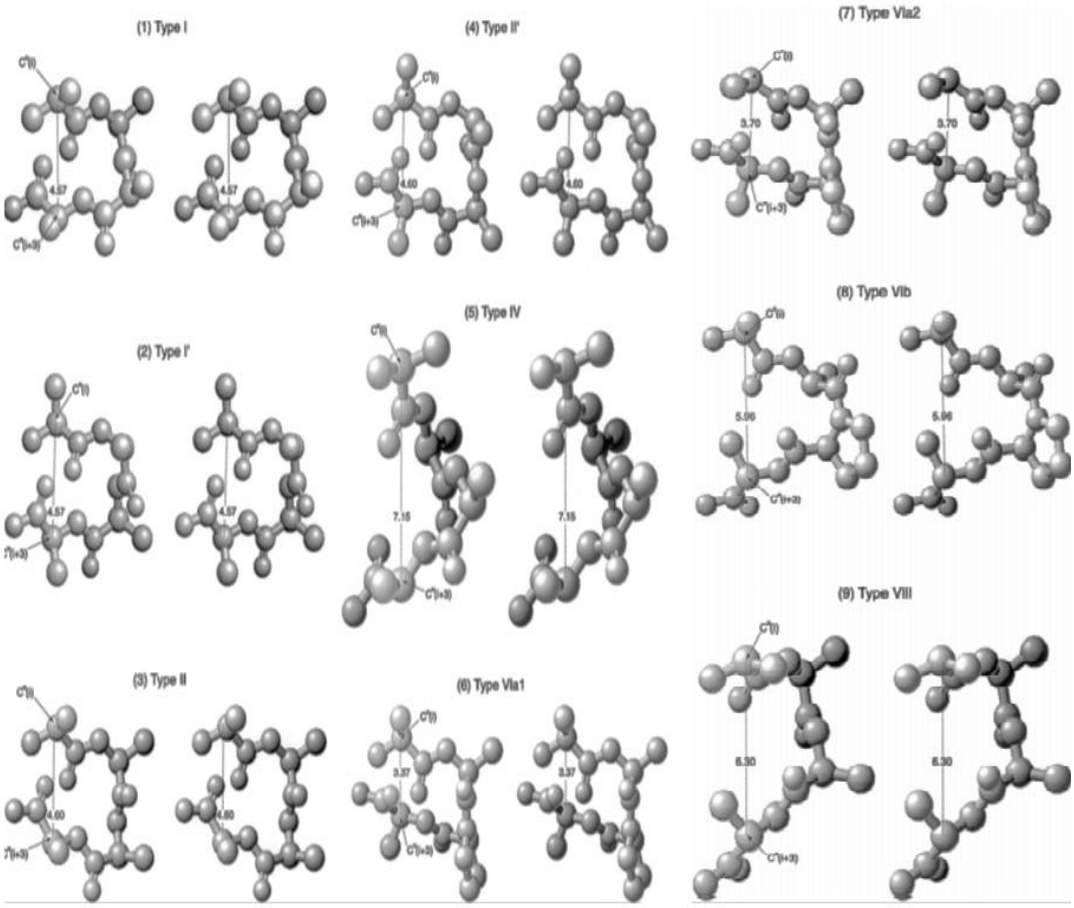


Figure 1.9 Illustrative stereo drawings of beta-turn types.

The distances between $C_\alpha(i)$-$C_\alpha(i+3)$ in type IV are slightly greater than 7Å since this type is a miscellaneous category and not really considered as an authentic β-turn [21]

.

1.1.4 Class-imbalance problems

In recent years, class-imbalance problems have been receiving many deep concerns because of their importance. A dataset is imbalanced if the number of samples in some classes is significantly larger than in other classes. In the case of two-class datasets, the class with small amount of samples is the minority (positive) class while the other is the majority (negative) class. For multi-class imbalanced datasets, there can be some minority classes, and in some situations, every class is the minority. However, in this thesis, we just focus on the two-class problem to agree with the common practices [46–50]. Figure 1.10 presents an illustration of imbalanced dataset.The class-imbalance problem is often found in the real decision systems which try to detect the rare but important cases such as fraud detection [51, 52], oil spills in

satellite images of the sea surface [53], risk management [54], text categorization [55] and so on. In the field of bioinformatics, this problem is very common, such as miRNA prediction [56], beta-turns prediction [33, 42], prediction of protein-interaction sites [16, 57, 58], protein-ATP binding residues prediction [59], microRNAs classification [60–62], translation initiation site recognition [63], et cetera. In some cases, the ratio of minority class to majority class can be as extreme as 1:100 or 1:100,000 [46]. When applying standard machine learning to the such datasets, it often harvests a poor performance that results from the accuracy. Most of the learning systems can be seriously influenced and tend to predict majority class exactly while users desire for both high sensitivity and specificity. One of the most common examples in real biomedical applications is the "Mammography Data Set," the collection of images acquired from a series of mammography exams performed on a set of distinct patients. Analyzing the images in a binary sense, the natural classes are labeled "Positive" for an image representative of a "healthy", and "Negative" for a "cancerous" patient. This data set contains 10,923 "Negative" samples and 260 "Positive" samples. We expect a classifier will provide 100% of predictive accuracy for both the minority and majority classes on the dataset. However, the reality showed that classifiers tend to provide a severe imbalanced degree of accuracy, with the majority class having close to 100% accuracy and the minority class having accuracies of 0-10 percent. If a classifier achieves 10% accuracy on the minority class of the mammography data set, it means that 234 minority samples are misclassified as majority samples. The consequence of this is equivalent to 234 cancerous patients diagnosed as noncancerous. This is clearly an undesired result [46].

In addition, class distribution and error costs also affect the learning algorithms. Standard classifiers assume that (i) the algorithms will perform on data drawn from the same distribution as the training data while the training and testing distributions are often different; (ii) the errors coming from different classes have the same costs while they are unlike in practice [64].

To solve this problem, many strategies have been proposed. Basically, all of them are divided into two categories: data level including the resampling methods, and algorithmic level including the methods aiming at adjusting the parameters of machine learning algorithms [46, 49]. However, [46] shows that resampling techniques are more effective on improving classifier accuracy than algorithm level

methods. Due to that reason, in this study, we mainly focus on the resampling techniques.

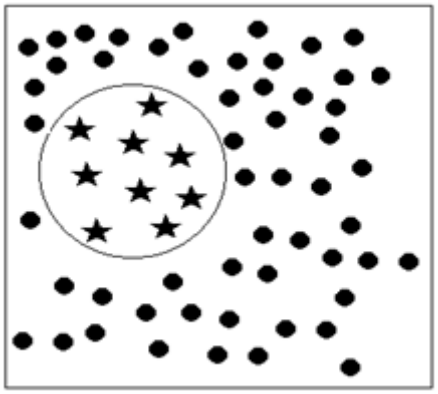


Figure 1.10 An illustration of an imbalanced dataset.

Blackened shapes represent samples; circles are majority class samples and stars are minority class samples**.**

1.2 Objectives

Because of the importance of predicting the interface residue and β-turn and what kind of turn it is, our thesis aims to the following problems:

Firstly, we would like to improve the performance of predicting protein interface residue by solving the problem of class-imbalance. To do that, we propose a new over-sampling algorithm for balancing the dataset. We chose the dataset that contains 2,829 interacting residues and 24,616 non-interacting residues for training and testing the predictor, and compare our results with the state-of-the-art approaches. We also combine our algorithm with some other methods to enhance the better results.

In addition, we try to use a new kind of feature for well distinguishing the protein interface and non-interface residues. We apply our new algorithm to this new dataset to evaluate the performance.

Secondly, we would like to better the quality of predicting β-turn . Since the high proportion of non-β-turn residues to the β-turn residues is one of the reasons decreasing the prediction's performance, we utilize random under-sampling method to balance the dataset. We create the well-characterized datasets for training and testing the model. We also apply this idea for predicting β-turn types. The results are compared with other state-of-the-art methods to evaluate the improvement.

1.3 Contributions

The main contributions of this thesis are described as below:

A novel over-sampling technique for relaxing the class-imbalance problem based on local density distributions. In order to alleviate the problem of overlapping and over-fitting simultaneously, we propose a novel over-sampling algorithm, which we name Over-sampling based on local Density (OSD). OSD algorithm focuses on only minority samples located where the local density of minority samples is small in comparison with that of majority samples. As the local minority density is smaller, OSD increases the number of minority samples more strongly by synthesizing artificial minority samples.

The enhancement on the performance of predicting protein-protein interaction sites by using our new over-sampling method OSD. We also proposed the methods combined with KSVM-THR and random under-sampling methods to reinforce the tolerance for the class imbalance problem. Results from experiments showed that the combination of our OSD algorithm and new feature group led to high sensitivity, precision, G-mean, MCC, F-measure, and AUC-PR, and comparable performance with the state-of-the-art methods. In addition, we found that the information of predicted shape strings increased the performance for predicting whether interface or non-interface residues.

The improvement in the performance of predicting β-turns and their types. We utilize predicted protein blocks and position specific scoring matrix together with random under-sampling method to improve the predicting the β-turns and their types. We executed the experiments on three benchmark datasets, and achieved MCCs of 0.58, 0.59 and 0.58 on the three datasets BT426, BT547 and BT823, respectively, in comparison of the state-of-art β-turn prediction methods. In the field of β-turn types prediction, we also harvested the high and stable results.

1.4 Thesis Organization

This thesis includes five chapters.

The first chapter is the current one that gives the basic concepts such as protein structure levels, protein blocks and the brief introduction of our research topic, thesis contributions and organization.

Chapter 2 introduces the overview of techniques for dealing with class-imbalance problems and evaluation metrics for imbalanced datasets classification.

Chapter 3 describes the improvement in predicting protein-protein interaction sites by using a novel over-sampling method and predicted shape strings.

Chapter 4 presents the improvement in the prediction of β-turns and their types applying predicted protein blocks and under-sampling method.

Chapter 5 concludes this thesis and mentions the future works.

Chapter 2

Methods for Dealing with Class-imbalance Problems

The methods to handle the class-imbalance problem are categorized into two groups, data-level methods and algorithm-level methods. This chapter aims to present briefly the methods that have been used to deal with this problem. Then, the performance evaluation measures such as overall accuracy, G-mean, Mathews Correlation Coefficient, and so on, which are often utilized to evaluate the classification performance on the imbalanced datasets are presented.

2.1 Standard Classifier Modeling Algorithm

There are many basic well-known classifier learning algorithms such as K-nearest Neighbors [65], Decision trees (ID3 [66], C4.5 [67]), Back-propagation Neural Networks [68], Support Vector Machines [69], and so forth. Due to the limitation of space, in this thesis, we just focus on Support Vector Machines that are mainly used for our research.

Support Vector Machines (SVMs), a popular machine learning technique, which have been successfully applied to many real-world classification problems from various domains, were proposed by Vapnik.

The goal of the SVM learning algorithm is finding the optimal hyper-plane to separate the dataset into two classes, with the maximal margin. Here, margin is the minimal distance from the hyper-plane to the closest data points. The solution is based only on the support vectors, which are the data points at the margin. SVMs originally were for the linear binary classification problem. However, in many applications, the linear classifier cannot work well but the non-linear classifier. In these cases, the non-linear separated problem is transformed into a high dimensional feature space using a set of non-linear basis functions. An important property of SVMs is that it is not necessary to know the mapping function explicitly. A kernel representation by a kernel function can be used, instead. When perfect separation is not possible, slack variables are introduced for sample vectors to balance the trade-off between maximizing the width of the margin and minimizing the associated error [48].

SVMs are believed to be less affected by the class imbalance problem than other classification learning algorithms [70] since boundaries between classes are calculated based on the support vectors and the class sizes may not affect the class boundary too much. However, some weaknesses of SVMs when applying to the imbalanced datasets were reported. [71] showed that in this case, the separating hyper-plane of an SVM model can be skewed towards the minority class, therefore can degrade the performance of the model with respect to the minority class. Wu and Chang [72] reported when the dataset is unbalanced, the positive samples lie further from the ideal boundary result in the boundary skew. They also said that in this case, the ratio

of positive and negative support vectors would be imbalanced. However, the authors in [73] objected to this idea.

2.2 The State-of-the-art Solutions for Class-imbalance Problems

2.2.1 Resampling techniques

Generally, resampling techniques aim to balance the distribution of the dataset by some mechanisms. This group includes the methods such as over-sampling the minority class, under-sampling the majority class, and combinations of the above techniques.

Over-sampling

Over-sampling method tries to balance the data set by increasing the number of minority class samples.

The simplest way is named Random Over-sampling, which randomly chooses some minority samples, replicates them and then adds to the original dataset. However, this strategy can lead to the over-fitting since over-sampling simply appends duplicated samples to the original data set, multiple instances of certain samples become "tied" [49, 74]. In addition, in case of large data sets, the cost in time and memory of classifying phase will be increased [46, 49].

Related synthetic sampling, the Synthetic Minority Over-sampling TEchnique (SMOTE) [75] is a powerful method that has been successfully applied for many research [76]. SMOTE tries to overcome the over-fitting by generating synthetic samples between each minority class instance and its randomly selected nearest neighbors. The synthetic sample x_{new} of a minority sample x_i is created by

$$x_{new} = x_i + \delta * (x_i - x_n)$$

where δ is a random number in [0,1] and x_n is one of the k nearest neighbors of x_i. Figure 2.1 presents an example of SMOTE.

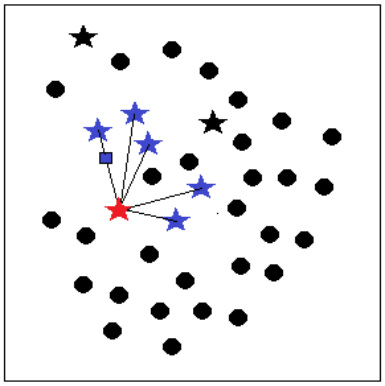


Figure 2.1 An illustration of SMOTE algorithm.

Dataset with majority class samples (circles) and minority class samples (stars). Minority sample x_i (in red) and its five nearest neighbors (in blue). The synthetic sample which is generated by x_i and one of its random chosen nearest neighbor is presented as the blue square.

Though SMOTE can overcome the drawback of Random Over-sampling, the numbers of synthetic samples corresponding to each minority class instance are the same may result in the overlapping between classes. Many improvements of SMOTE, therefore, were developed, such as SMOTEBoost [77], Smote-RSB [78], Safe-Level-SMOTE [79], Borderline-SMOTE [80] and so on.

The other over-sampling methods that need to pay attention to are the Cluster-based sampling algorithms. These methods are more flexible than the simple and synthetic sampling algorithms, and can be tailored to target very specific problems. CBO, the cluster-based over-sampling algorithm [81], effectively deals with the within-class imbalance problem [46]. The basic idea of this method is clustering before over-sampling. Specifically, in [81], the authors used K-mean to cluster the whole dataset. Then, both the minority class and majority class were oversampled. All the clusters in the majority class were randomly oversampled but the largest one. After this step, every majority cluster had the same size. In the minority class, each cluster was oversampled so that it would contain *maxsize/nclusters* samples, where *maxsize* was the overall size of the majority class after over-sampling, and *nclusters* was the number of minority clusters. The illustrative example of this method is in Figure 2.2 [46].

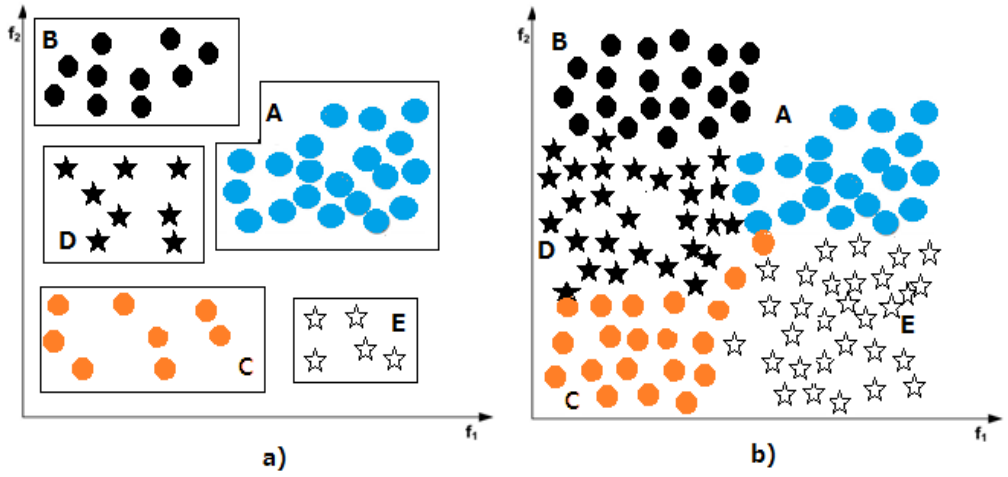


Figure 2.2 Cluster-Based Sampling method example.

a) Dataset with three majority clusters (A, B, C) and two minority clusters (D, E). Cluster A contains the most number of samples.

b) After applying the method, every cluster contains the same number of samples as cluster A.

Under-sampling

Contrary to Over-sampling, Under-sampling method solves the class-imbalance problem by decreasing the number of majority class samples, therefore, decreases the cost of computation.

Random Under-sampling balances the original data set distribution by randomly eliminating some majority samples. However, this way may lead to lose a lot of important information of the majority class.

EasyEnsemble, BalanceCascade [82] were proposed to overcome this limitation. EasyEnsemble develops an ensemble learning system by independently sampling several subsets from the majority class and developing multiple classifiers based on the combination of each subset with the minority class samples. On the other hand, the BalanceCascade develops an ensemble of classifiers to select which majority class samples for under-sampling systematically.

The other under-sampling methods that based on k-nearest neighbors are NearMiss-1, NearMiss-2, Near-Miss-3, and the “most distant” method [50]. The NearMiss-1 method chooses majority samples whose average distance to the three minority class nearest neighbors is the smallest. The NearMiss-2 method selects the majority class samples whose average distance to the three farthest minority class

neighbors is the smallest. NearMiss-3 selects a given number of majority class samples that are closest to each minority sample to guarantee that every minoritysample is surrounded by some majority examples. The “most distance” method selects the majority class samples whose average distance to the three minority class nearest neighbors is the largest.

Anand et al. [61] introduced an under-sampling method that also based on nearest neighbor and weighted SVM. For each minority class sample, its k closest majority class samples will be removed. The distance between samples here is weighted Euclidean distance.

2.2.2 Algorithm level methods for handling imbalance

This group of methods modifies the standard classification algorithm to account for class-imbalance. A popular way for dealing with the class-imbalance problem is to choose a proper inductive bias. For decision trees, approaches are adjusting the probabilistic estimate at the tree leaf [83, 84] or developing new pruning techniques [83].

For SVMs, the use of different penalty constants for different classes (cost-sensitive) [73, 85, 86], and adjusting the class boundary based on kernel-alignment ideal [72] were proposed.

Cost-sensitive learning methods deal with the class-imbalance problem by considering the costs associated with misclassifying samples [87, 88]. One of the simple ways is adjusting the decision threshold in assigning class memberships. Chen et al. [89] shows that the adjustment decision threshold can increase the sensitivity and decrease specificity via the experiments on for four classification algorithms: logistic regression model, classification tree, Fisher’s linear discriminant and modified nearest neighbor. Using the same idea, Lin and Chen [90] proposed the SVM-THR method that adjusted the decision threshold of SVM. These methods are said to be naturally applied to handle the imbalanced datasets [46].

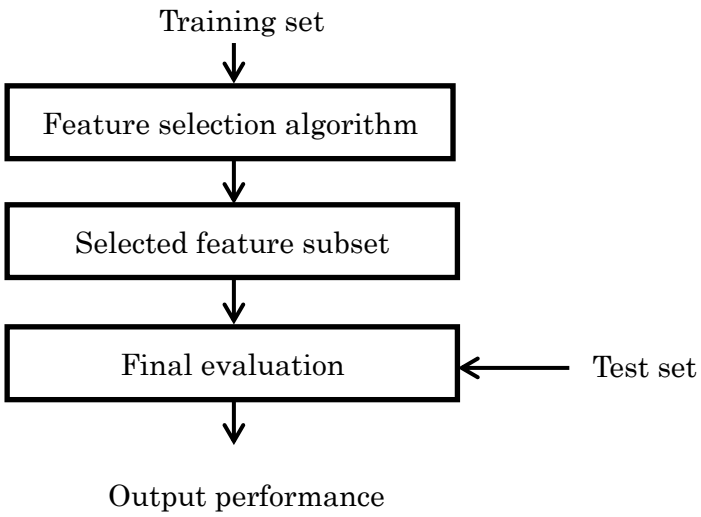
The other strategy is one-class learning method. The one-class learning approach learns on only one class to determine the decision boundary [91, 92]. Raskutti and Kowalczyk [93] demonstrates that one-class learning method performs well for extreme imbalanced datasets composed of a high dimensional noisy feature space.

One drawback of these methods is the requirement of the algorithm-specific modification.

2.3 Feature Selection for Imbalance Datasets

Feature selection is a pre-processing technique that to select a subset of best features. The purpose of feature selection is to avoid over-fitting and improve model's performance, to provide a cost-effective model, and to gain a deeper insight into the underlying processes that generated the data [94]. In the field of imbalanced datasets mining, feature selection is even more important than the choice of the learning method [64, 95].

The general feature selection process is described as follow:



A feature selection algorithm belongs to one of three groups: filter methods, wrapper methods, and embed methods.

Filter method selects the features based on their relevance scores. The relevance scores of features are calculated by various feature-ranking techniques such as Euclidean distance, Chi-squared, Information Gain, Gain Ratio, Symmetric Uncertainty, ReliefF, and so on [96]. These methods are fast, easily scale for high dimensional datasets, independent of the classification algorithm but ignoring the

interaction with classifier [94]. The general scheme of filter method description is in Figure 2.3.

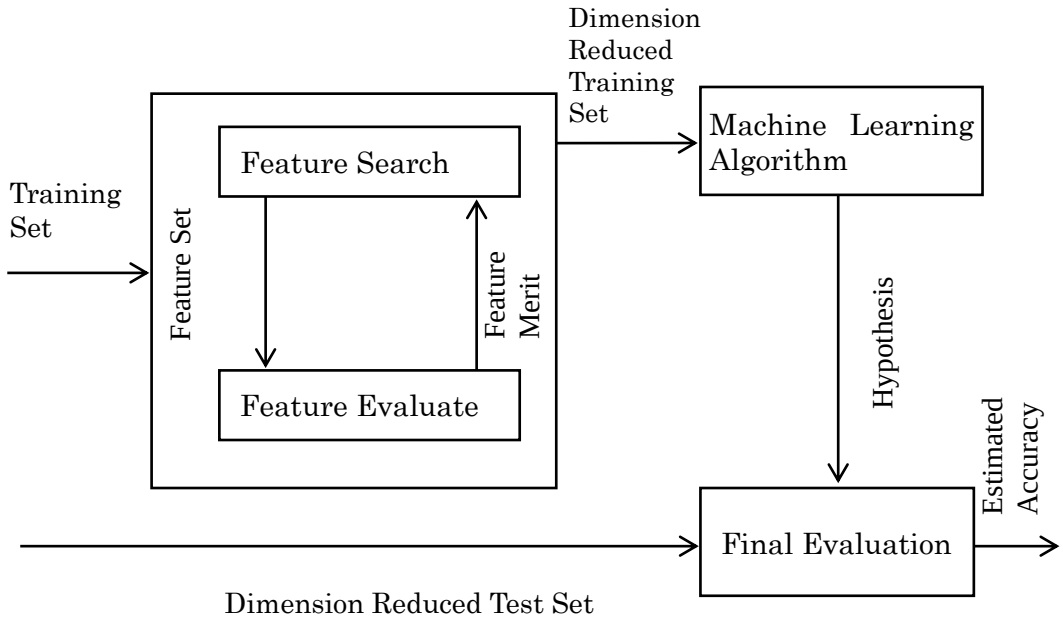


Figure 2.3 Filter method. Figure adapted from [97]

Wrapper methods (Figure 2.4), such as Sequential forward selection technique, Sequential backward selection technique, SVM-RFE, ect., use the classifier to calculate the score of feature-subsets based on their predictive power. These methods pay attention to the feature dependencies and interact with the classifier. However, the common drawback is that they are computationally intensive and have high risk of overfitting [94, 97].

The embedded methods can be seen as the hybrid methods with the combination of filter and wrapper methods. Firstly, filter model is applied to identify the goodness of features. Then, a wrapper model is performed to choose the optimal feature-subset. Table 2.1 from [94] presents the taxonomy of feature selection techniques.

Table 2.1 A taxonomy of feature selection techniques [94]

Model search	Advantages	Disadvantages	Examples
Filter (Uni-variate)	• Fast • Scalable • Independent of the classifier	• Ignores feature dependencies • Ignore interation with the classifier	• Chi-squared • Euclidean distance • t-test • Information gain, Gain ratio
Filter (Multivariate)	• Models feature dependencies • Independent of the classifier • Better computational complexity than wrapper methods	• Slower than univariate techniques • Less scalable than univariate techniques • Ignores interaction with the classifier	• Correlation-based feature selection (CFS) • Markov blanket filter • Fast correlation-based feature selection (FCBF)
Wrapper (Deterministic)	• Simple • Interacts with the classifier • Models feature dependencies • Less computationally • Intensive than randomized methods	• Risk of over fitting • More prone than randomized algorithms to getting stuck in a local optimum (greedy search) • Classifier dependent selection	• Sequential forward selection (SFS) • Sequential backward elimination (SBE) • Plus q take-away r • Beam search
Wrapper (Randomized)	• Less prone to local optima • Interacts with the classifier • Models feature dependencies	• Computationally intensive • Classifier dependent selection • Higher risk of over-fitting than deterministic algorithms	• Simulated annealing • Randomized hill climbing • Genetic algorithms • Estimation of distribution algorithms
Embedded	• Interacts with the classifier • Better computational complexity than wrapper methods • Models feature dependencies	• Classifier dependent selection	• Decision trees • Weighted naïve Bayes • Feature selection using the weight vector of SVM

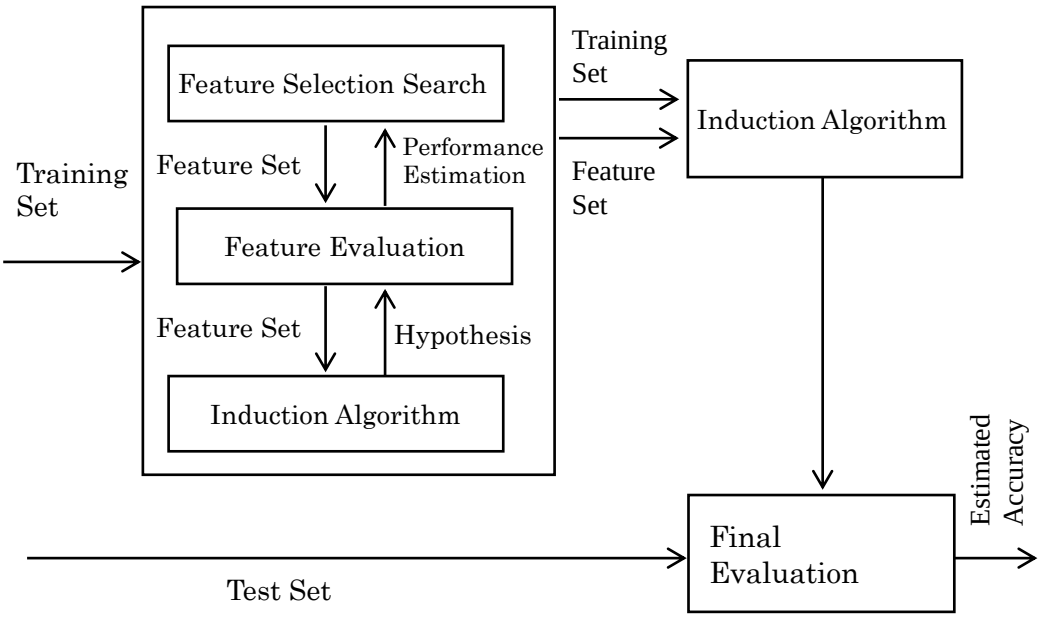


Figure 2.4 Wrapper method. Figure adapted from [98]

2.4 Evaluation Metrics

Evaluation measures aim to evaluate the classification performance and to guide the classifier modeling. For the normal situation, overall accuracy is often used. However, when performing the classification on the imbalanced datasets, overall accuracy is no longer suitable for evaluating the performance of classifier [99]. If the class-imbalance problem is severe, a naive approach will make the overall accuracy very high even though, most samples are assigned to the majority class and no sample is assigned to the minority class [46].

Thus, besides overall accuracy, in this study, the other metrics such as sensitivity, specificity, G-mean, F-measure and Matthews correlation coefficient are used, which are defined as follows:

$$\text{Overall accuracy} = (TP + TN)/(TP + FN + TN + FP)$$

$$\text{Sensitivity} = \text{Recall} = \text{TP}/(\text{TP} + \text{FN})$$

$$\text{Specificity} = \text{TN}/(\text{TN} + \text{FP})$$

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP})$$

$$\text{G-mean (Balanced accuracy)} = \left(\frac{\text{TP}\times\text{TN}}{(\text{TP}+\text{FN})\times(\text{TN}+\text{FP})}\right)^{1/2}$$

$$\text{F-measure} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}}$$

$$\text{Matthews Correlation Coefficient (MCC)} = \frac{\text{TP}\times\text{TN}-\text{FP}\times\text{FN}}{\left((\text{TP}+\text{FN})\times(\text{TP}+\text{FP})\times(\text{TN}+\text{FP})\times(\text{TN}+\text{FN})\right)^{1/2}}$$

where TP is the number of positive samples that are correctly predicted as positive; TN is the number of negative samples that are correctly predicted as negative; FP is the number of negative samples that are predicted as positive; and FN is the number of positive samples that are predicted as negative.

Sensitivity and specificity have been commonly used in medical community [27]. G-mean is the combination of both sensitivity and specificity [24]. F-measure is the harmonic mean of precision and recall. Matthews correlation coefficient measures how good the correlation of the predicted class labels and the actual class labels is. It lies in the range from -1 to 1, where -1,1, and 0 represents the worst, the best and the random predictor, respectively.

In addition, the threshold independent measures ROC (Receiver Operating Characteristics) curve and AUC (Area Under the Curve), which are often used in bioinformatics [100], are adopted. ROC graphs are two-dimensional graphs with the Y axis and the X axis are TP rate and FP rate, respectively. An ROC graph pictures relative tradeoffs between true positives (benefits) and false positives (costs). From ROC graph, AUC can be calculated. The AUC of a classifier represents the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance [101]. AUC receives the value between 0 and 1. An acceptable classification model should have AUC above 0.5. An AUC value above 0.7 indicates a useful prediction, and a good prediction method achieves AUC above 0.85.

Chapter 3

Improving the Prediction of Protein-Protein Interaction Sites Using a Novel Over-sampling Approach and Predicted Shape Strings

Identification of protein-protein interaction (PPI) sites is one of the most challenging tasks in bioinformatics and many computational methods based on support vector machines have been developed. However, current methods often fail to predict PPI sites mainly because of the severe imbalance between the numbers of interface and non-interface residues. In this study, we propose a novel over-sampling method that relaxes the class-imbalance problem based on local density distributions.We applied the proposed method to a PPI dataset that includes 2,829 interface and 24,616 non-interface residues. The experimental result showed a significant improvement in predictive performance comparing with the other state-of-the-art methods according to the six evaluation measures.

3.1 Introduction

Protein-protein interactions, known as physical contacts among proteins, are essential molecular processes for living organisms to maintain their lives. They play a central role in various biological functions such as regulation of metabolic and signaling pathways, DNA replication, protein synthesis, immunological recognition, and so forth. Especially, physical interface between two interacting proteins is a key to understand enzymatic activities of proteins. Therefore, one important task in bioinformatics is to develop computational methods to find binding interfaces between two interacting proteins accurately.

However, a naive approach based on support vector machines, one of the most standard classifiers, often fails to predict binding interfaces among interacting proteins with high specificity since the number of non-interaction residues is much larger than the number of interaction residues. This is so-called the class-imbalance problem. A dataset is imbalanced if the number of samples in some classes is significantly larger than in other classes. In the serious cases, the ratio of minority class to majority class can be as large as 1:100,000 [46]. Use of traditional machine learning techniques for these datasets often leads to undesirable results that only majority class is correctly predicted. This is a common problem in bioinformatics such as prediction and classification for miRNAs [56], beta-turns [33, 42], microRNAs [60, 61], breast cancer, lung cancer [90] and so on.

Many methods to deal with the class-imbalance problem have been developed. One important class of such methods is resampling-based techniques such as over-sampling and under-sampling methods, which have been reported to improve classification accuracy significantly [46]. In this study, we propose a novel over-sampling approach in order to relax class-imbalance for the dataset of PPI sites. Instead of dealing with all minority class samples equivalently, we intentionally increase the number of minority samples according to their local distribution. Furthermore, predicted shape strings, which have been utilized in many researches in recent years [102–104], are used to enrich the feature groups. We present numerical experiments compared with state-of-the-art methods such as Anand et al. [61].

3.2 Materials and Methods

3.2.1 Dataset

In this study, we used two datasets. The first one (that was named D1050) was the same with Chen and Jeong [16]. For predicting interface residues and non-interface residues, Chen and Jeong used the information of physicochemical features, evolutionary conservation score, amino acid distances, and position specific score matrix (PSSM) to extract features for 99 polypeptide chains of 54 hetero complexes [11]. By using a sliding window with size 21, the central residue of a partial peptide was assigned as interface residue if its relative solvent accessible surface area (RASA) was greater than 25% and the difference of accessible surface areas (ASAs) between its unbound state and bound state was greater than $1Å^2$. As a result, each residue was represented as a 1,050 features. The dataset contained 2,829 interface residues (positive class) and 24,616 non-interface residues (negative class). The ratio of positive class samples to negative class samples was 1:8.7. That is, this dataset was highly imbalanced.

The second dataset (was named D1239) was prepared by adding information of predicted shape strings to the original dataset. The shape string of a protein is a sequence of symbols categorized according to the phi-psi torsion angles. There are eight shape symbols representing for eight categories (S,R,U,V,K,A,T,G). DSP program [104] was used to predict the shape strings. Each residue was predicted as one of these eight states or state N as the undefined phi-psi angle pair. Each sample in this dataset includes 1, 239 features.

3.2.2 Methods

Resampling techniques

As presented in [46], resampling techniques such as over-sampling methods, under-sampling methods, and under-over-sampling combination methods effectively improve classification accuracy for imbalanced datasets. Under-sampling methods balance the imbalanced dataset by removing samples in the majority class until the dataset becomes balanced. An important disadvantage of under-sampling methods is that this removal of majority samples leads to a significant information loss for the majority group. On the contrary, over-sampling methods increase the number of

samples in the minority class. The synthetic samples are generated by various methods. The most naive technique is random over-sampling, which arbitrarily chooses some minority samples and replicates them (one or many times). One of the other common methods is SMOTE [75], which synthesizes the new samples locating between each minority class sample and its randomly chosen nearest neighbors. While random over-sampling techniques often lead to the over-fitting, SMOTE may result in the overlapping between classes [46]. Especially when the number of minority samples is small and they are distributed sparsely among the majority samples, the problem becomes more serious because most of the synthetic samples will be located among the majority class samples. Prati et al. [105] showed that the decrease in classification performance is caused by not only class-imbalance but also data-overlapping. Borderline-SMOTE [80] addresses this drawback by generating new samples for minority samples if they are located near the borderline, while the samples, which are surrounded by majority samples or have enough minority nearest neighbors are not considered. Though Borderline-SMOTE successfully improved predictive accuracy for imbalanced datasets, the overlapping problem is not carefully avoided.

In order to alleviate the problem of overlapping and over-fitting simultaneously, we propose a novel over-sampling algorithm, which we call Over-sampling based on local Density (OSD). Instead of generating the same number of synthetic samples for each minority sample as SMOTE, OSD algorithm focuses on only minority samples located where the local density of minority samples is small in comparison with that of majority samples. As the local minority density is smaller, OSD increases the number of minority samples more strongly by synthesizing artificial minority samples. Here we define local density for each sample as follows:

Definition 1. Suppose m and n are the numbers of samples with the same and different class labels for sample x, respectively. Local density of x with radius r is the proportion m/(m+n).

OSD- a novel over-sampling approach

A key idea of the OSD algorithm is to increase the number of minority samples located where the local density of minority samples is small in comparison with

majority samples. For each minority class sample *x*, first of all, OSD finds neighbors of *x* and divides into two groups, majority and minority neighbors, according to their class labels (line 2). Note that the terms "majority" and "minority" are used in the global context. Here, neighbors of *x* are defined as samples in hyper-sphere with radius *r*. The number of synthetic samples for each *x* depends on its local distribution with parameter *d* (lines 6-9):

- If *x* doesn't have neighbor (i.e. $m + n = 0$), or local density of *x* is 0 (i.e. $m = 0$), *x* locates far from the other minority samples and OSD generates the maximum number of synthetic samples with the same class labels as *x* in order to avoid the class imbalance problem and diminish boundary variance derived from local sparsity, simultaneously. Hence, *d* new samples will be synthesized.
- If local density of *x* is greater than 0, $d*(1-m/(m+n))$ new synthetic samples are created.
- If sample *x* has no different class label neighbor, OSD does not adjust the local density of *x*.

Then, OSD generates the samples by function **New_sample_generation** (line 10). The synthesized samples are generated so that their distances to *x* are always less than *r_min* and they tend to be located closer to x as follows: (1) OSD randomly generates a number r' which follow the density $p(r) = cr^{-1/2}$(0<r<1) where $c=r\_min\ /k^{1/2}$ with k is the number of features. (2) adds it to the element of feature vector (lines 14-15). The pseudo-code for OSD algorithm is as follows:

OSD algorithm

Input: Minority dataset *M*; Majority dataset *N*; ratio of generation *d*; radius *r*.
Output: Set of synthetic samples.
Begin
1. For each $x \in M$
2. calculate the local minority neighbors *m* & local majority neighbors *n* for *x*;
3. calculate the distance *r_min* from *x* to its local majority nearest neighbor;

4. if (*r_min*>*r*)
5. *r_min* = *r*;
6. if (*m*+*n* = 0)
7. *number_of_new_samples* = *d*;
8. else
9. *number_of_new_samples* = *d**(1-*m*/(*m*+*n*));
10. **New_samples_generation**(*x*,*r_min*, *number_of_new_samples*);
11. End_for

End

Function New_sample_generation(x, r_min, d)

Input: Sample $x = (x_1,x_2,\ldots,x_k,class\_label)$; number of new samples *d*; radius *r_min.*

Output: Set of synthetic samples *new_samples_array* of *x*.

Begin

12. For *i* = 1:*d*
13. *new_sample_class_label* = *class_label*;
14. for *j* = 1:*k*
15. $new\_sample_j = x_j + r'$;
16. end_for
17. push(*new_samples_array*,*new_sample*);
18. end_for

End

KSVM-THR

We note that OSD generally does not balance imbalanced datasets entirely. To address this issue, we combine OSD and KSVM-THR, SVM with adjustment of the decision parameter, proposed by Lin and Chen [90]. The decision threshold θ of KSVM-THR is defined as

$$\theta = -1 + 2 * (p + \alpha)/(p + n + 2 * \alpha)$$

where p and n are the numbers of minority and majority class samples, respectively.

The constant α is the tuning parameter and in the experiments below, it was optimized by grid search. If a data set is balanced, θ becomes zero. In this study, we utilize this technique to compose OSD-THR and RU-OSD-THR that combine KSVM-THR with OSD and RUS-OSD (Random Under-sampling –OSD).

Experimental design

SVM with Gaussian RBF kernel was utilized to create a basic classifier. We conducted 10-fold cross validation. All the features of the datasets were normalized. Noise samples in the datasets were filtered out before over-sampling, where we defined samples that have the same feature vector and that belongs to different classes as noise samples. The overall predicting process is shown in Figure 3.1. To determine the radius *r* for algorithm OSD, we calculated the distance between each pair of samples in the training set, sorted them in ascending order, saved in array *D*, set *k* = dim(*D*)*0.1% (*k* = dim(*D*)*0.01% for the D1239) where dim(*D*) was the size of *D* and assigned *r* as value of element k^{th} of *D*.

Since the ratio of positive class to negative class of this dataset is 1:8.7, overall accuracy is not suitable for evaluating the performance of classifier. If the class-imbalance problem is severe, a naive approach that assigns all samples to the majority class makes overall accuracy high though no sample was assigned to the minority class [46]. Thus, as measures of performance evaluation, we use overall accuracy, sensitivity, specificity, G-mean and Matthews correlation coefficient, which are defined as follows:

$$\text{Overall accuracy} = (TP + TN)/(TP + FN + TN + FP)$$

$$\text{Sensitivity} = TP/(TP + FN)$$

$$\text{Specificity} = TN/(TN + FP)$$

$$\text{G-mean (Balanced accuracy)} = (Sensitivity \times Specificity)^{1/2}$$

$$\text{Matthews correlation coefficient } (MCC) = \frac{\mathrm{TP}\times\mathrm{TN}-\mathrm{FP}\times\mathrm{FN}}{\left((\mathrm{TP}+\mathrm{FN})\times(\mathrm{TP}+\mathrm{FP})\times(\mathrm{TN}+\mathrm{FP})\times(\mathrm{TN}+\mathrm{FN})\right)^{1/2}}$$

Where TP and TN are the numbers of interface residues and non-interface residues that are correctly predicted; FP and FN are the numbers of non-interface residues and interface residues that are predicted as different from what they really are. Sensitivity

and specificity have been commonly used in medical community [61]. G-mean is the combination of both sensitivity and specificity [15]. Matthews correlation coefficient measures how good the correlation of the predicted class labels and the actual class labels is. It lies in [-1,+1], where -1,1, and 0 represents the worst, the best and the random predictor, respectively.

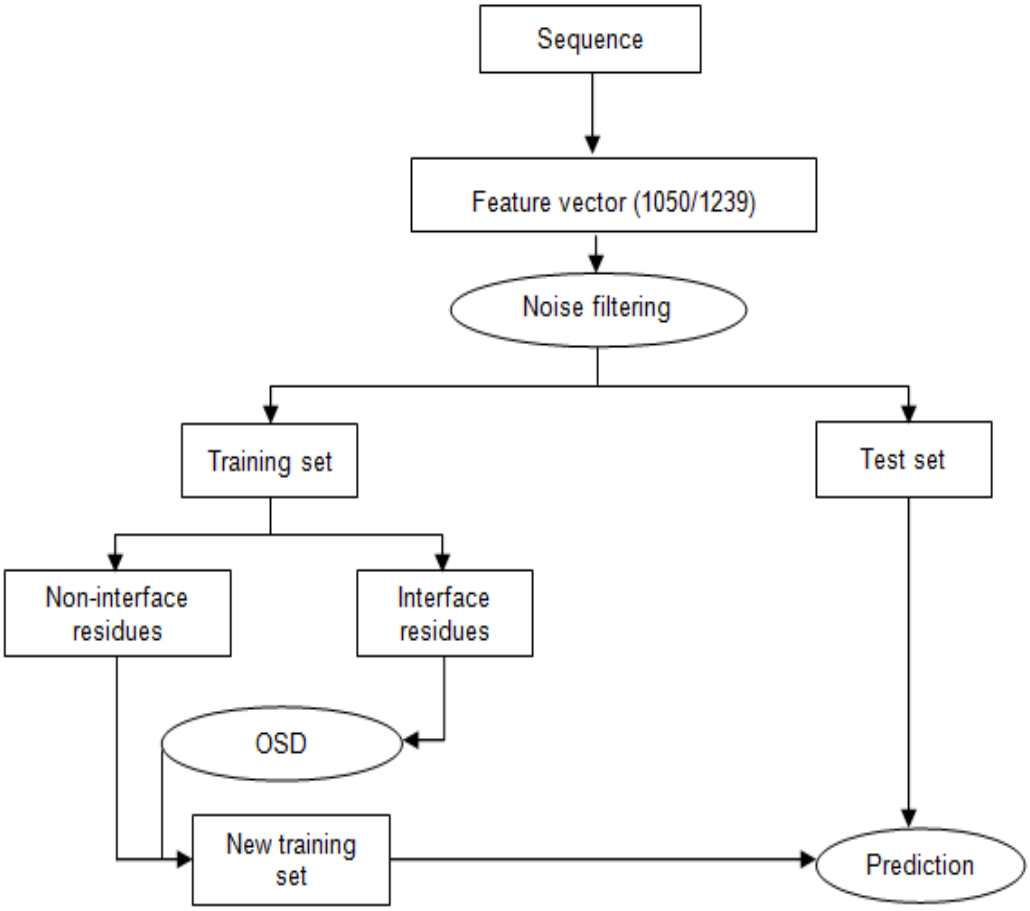


Figure 3.1 Schematic representation of our method

3.3 Results and Discussions

3.3.1 Evaluation on the D1050 Dataset

Using D1050 dataset, we evaluated the performance of OSD algorithm. It was

compared with KSVM without resampling (KSVM-only), Random Under-sampling (RUS), KSVM-THR-only, weighted SVM, SMOTE, the method of Chen and Jeong, and the under-sampling method introduced by Anand et al. [61]. The results of all these methods are shown in Table 3.1. In addition, Table 3.2 shows the results of experiments with the different decision thresholds of the methods.

Since non-interface residues approximately nine times outnumbered interface residues, KSVM-only could not perform well, whereas weighted-SVM, which assigns different costs of misclassification to minority and majority classes, could predict more positive samples than KSVM-only. Also, KSVM-THR-only achieved better performance by decreasing the decision threshold.

RUS removed many negative samples to balance the dataset (the new ratio of negative: positive samples was 1.1:1) so it improved the prediction results in comparison with KSVM-only and weighted-SVM but the best previous method (Anand et al.). However, RUS-THR was worse than RUS: since RUS itself balanced the dataset, the decrease in the decision threshold resulted in a higher sensitivity and low specificity. Meanwhile, RUS-OSD achieved better sensitivity, specificity, and G-mean than the corresponding results of Anand et al. by eliminating a part of majority class samples and then using OSD to increase the minority class samples.

Two of our over-sampling methods, OSD and OSD-THR, outperformed the method of Anand et al. (Table 3.1). For example, overall accuracy, specificity, and G-mean of OSD were 10.70%, 12.30%, and 3.36% higher than the competing method while sensitivity was 3.18% lower. The latter approach, OSD-THR, was better than the best previous method at all evaluation metrics.

Since MCC was not reported in [61], we could not directly compare with their method, under various conditions. However, at least under the condition that sensitivity equals to 70%, the MCC values of the method in [16] and our method were 0.32 and 0.48, respectively. Figure 3.2 describes the correspondence between MCC and sensitivity of KSVM-only and OSD.

Figure 3.3 demonstrates the ROC curves of OSD and the other methods. ROC curve of Cheng and Jeong was taken from [16]. It shows that while RUS decreased the performance of KSVM-only, the combination of RUS and OSD achieved a better result.

Table 3.1 Performance measures comparison of different methods on the dataset D1050 in terms of best G-mean

Method	Overall accuracy (%)	Sensitivity (%)	Specificity (%)	G-mean
KSVM-only	90.11	4.66	99.93	21.59
OSD	88.23	67.86	90.57	78.40
RUS (1.1:1)	76.17	70.59	76.81	73.63
RUS-OSD	75.31	80.73	74.69	77.65
KSVM-THR-only	90.66	11.48	99.76	33.85
OSD-THR	83.36	77.73	84.01	80.80
RUS-THR(1.1:1)	65.71	82.11	83.82	72.39
RUS-OSD-THR	64.94	88.51	62.24	74.22
Weighted-SVM*	91.57	55.87	95.56	73.08
SMOTE*	92.96	51.74	97.69	71.07
Chen and Jeong (2009)*	71.90	71.20	71.98	71.59
Anand et al. (2010)*	77.53	71.04	78.27	74.54

**: Result was taken from the paper of Anand et al.*

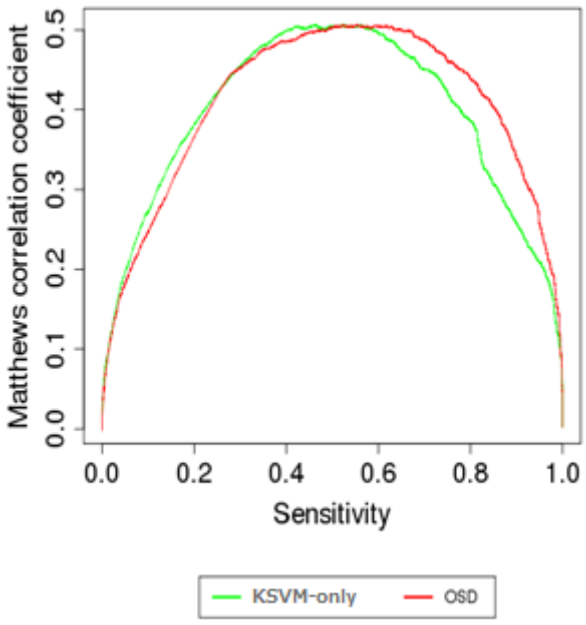


Figure 3.2 MCC vs. sensitivity of the two methods KSVM-only and OSD on the D1050 dataset

Table 3.2 Performance of KSVM-THR-only, OSD-THR, RUS-THR and RUS-OSD-THR with different decision threshold values on the dataset D1050

Method	**KSVM-THR-only**				**OSD-THR**				**RUS-THR**				**RUS-OSD-THR**			
Thr	**ACC**	**SN**	**SP**	**G**	**ACC**	**SN**	**SP**	**G**	**ACC**	**SN**	**SP**	**G**	**ACC**	**SN**	**SP**	**G**
0.96	89.69	0.07	1.00	2.65	91.93	31.07	98.93	55.44	90.34	21.63	98.24	46.10	90.66	31.28	97.49	55.22
1.73	89.69	0.00	1.00	0.00	89.68	0.00	99.99	0.00	89.71	0.42	99.97	6.51	89.72	0.42	99.98	6.51
8.52	89.69	0.00	1.00	0.00	89.69	0.00	1.00	0.00	89.69	0.00	1.00	0.00	89.69	0.00	1.00	0.00
-2.92	10.30	1.00	0.00	0.00	10.31	1.00	0.00	0.00	10.31	1.00	0.00	0.00	10.30	1.00	0.00	0.00
-1.24	17.60	99.71	8.16	28.54	35.80	98.23	28.63	53.03	20.02	99.46	10.89	32.91	21.19	99.78	12.16	34.83
-0.85	90.23	58.71	93.85	74.23	62.94	91.69	59.64	73.94	38.16	96.50	31.45	55.09	39.08	97.84	32.32	56.24
-0.79	91.52	49.80	96.32	69.26	65.90	90.70	63.05	75.62	41.35	95.72	35.10	57.97	41.97	97.13	35.63	58.83
-0.73	92.03	43.51	97.61	65.17	68.36	89.74	65.90	76.90	43.90	94.76	38.06	60.05	44.43	96.42	38.45	60.89
-0.58	91.91	29.26	99.11	53.85	74.74	86.63	73.37	79.73	51.83	91.19	47.31	65.68	51.82	94.23	46.94	66.51
-0.45	91.34	20.25	99.51	44.89	78.72	83.28	78.20	80.70	57.81	87.84	54.35	69.10	57.28	92.08	53.28	70.04
-0.37	91.01	15.69	99.67	39.55	81.14	80.98	81.16	81.07	61.57	85.25	58.85	70.83	60.83	90.77	57.39	72.18
-0.32	90.81	13.22	99.73	36.31	82.47	79.28	82.84	81.04	63.92	83.42	61.67	71.73	63.14	89.46	60.11	73.33
-0.28	90.66	11.48	99.76	33.85	83.36	77.73	84.01	80.80	65.71	82.11	63.82	72.39	64.94	88.51	62.24	74.22

**Thr = Decision threshold; *ACC = accuracy (%); *SN = sensitivity (%); *SP = specificity (%); *G = G-mean (%)*

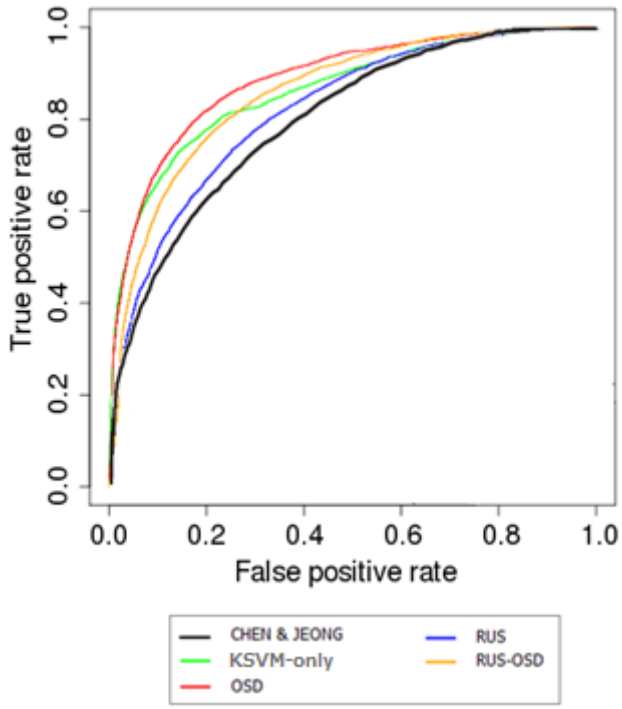


Figure 3.3 ROC curves of the competing methods on the D1050 dataset

3.3.2 Evaluation on the D1239 Dataset

We conducted experiments on the D1239 dataset and compared with the results of the D1050 to evaluate the effect of shape strings and the new over-sampling algorithm on the PPI sites prediction problem.

In addition to the evaluation criteria above, F-measure and Area Under Precision/Recall Curve (AUC-PR) [16] were used. F-measure is defined as follows:

$$\text{F-measure} = (2 * precision * recall)/(precision + recall)$$

where:

$$\text{precision} = TP/(TP + FP)$$

$$\text{recall} = TP/(TP + FN)$$

These metrics show the ability of classifier for detecting rare positive samples in the imbalanced dataset. Table 3.3 shows the results of experiments on the dataset D1239 with the different decision thresholds of the methods. Table 3.4 shows the improvements using our algorithm and new decision threshold in the comparison of the naïve classifier. In Table 3.4, OSD and OSD-THR outperformed the others and the best previous result in G-mean. It indicates that our over-sampling algorithm based on the local density can relieve the class-imbalance problem in this dataset. On the other hand, KSVM-only and KSVM-THR-only on the dataset D1239 achieved higher accuracy, sensitivity, G-mean than on the D1050. It demonstrated that shape string is an informative feature for discriminating interface and non-interface residues.Figure 3.4 and Figure 3.5 show that performance curves on D1239 are similar to the ones on D1050.

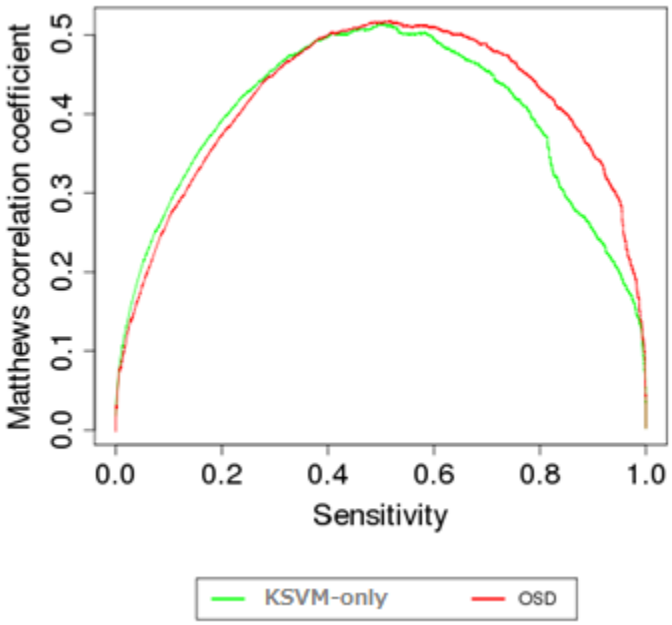


Figure 3.4 MCC vs. sensitivity of KSVM-only and OSD on the D1239 dataset

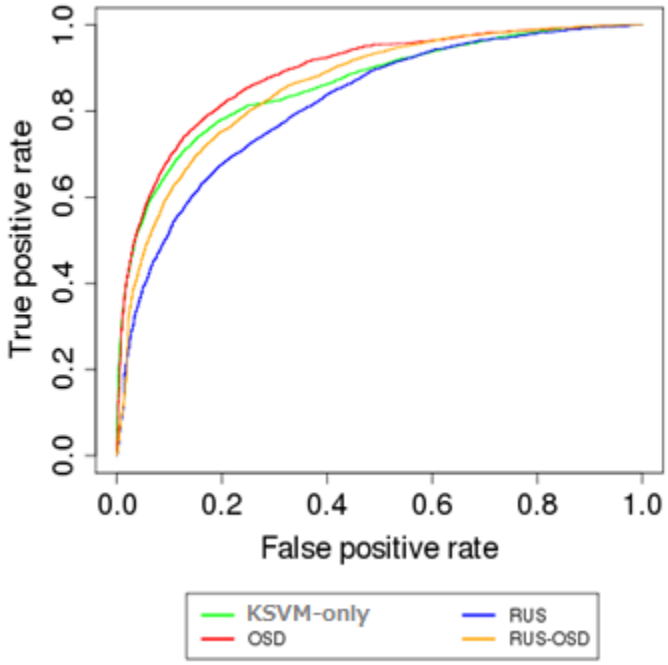


Figure 3.5 ROC curves of the competing methods on the D1239 dataset

Table 3.5 displays the comparative results on the datasets D1050 and D1239. Though sensitivity of OSD and OSD-THR decreased 4.73% and 3.29% (from 67.86% to 63.13% and from 77.73% to 74.44%), respectively, precision increased 4.45% and 3.42%. All the experiments on D1239 achieved higher F-measure than the corresponding one on the D1050. In addition, F-measure of OSD and OSD-THR on the both datasets are higher than that one of Chen and Jeong (49%) [17]. Furthermore, AUC-PR of KSVM-only and OSD on D1050 and D1239 were 0.56, 0.55, 0.58, and 0.57, respectively. In Figure 3.6, it can be seen that the performance of KSMV-only on D1239 is apparently better than the one on D1050 in the area of recall lower than 0.3 and precision higher than 0.8. It means that shape string is effective for performance improvement in this area.

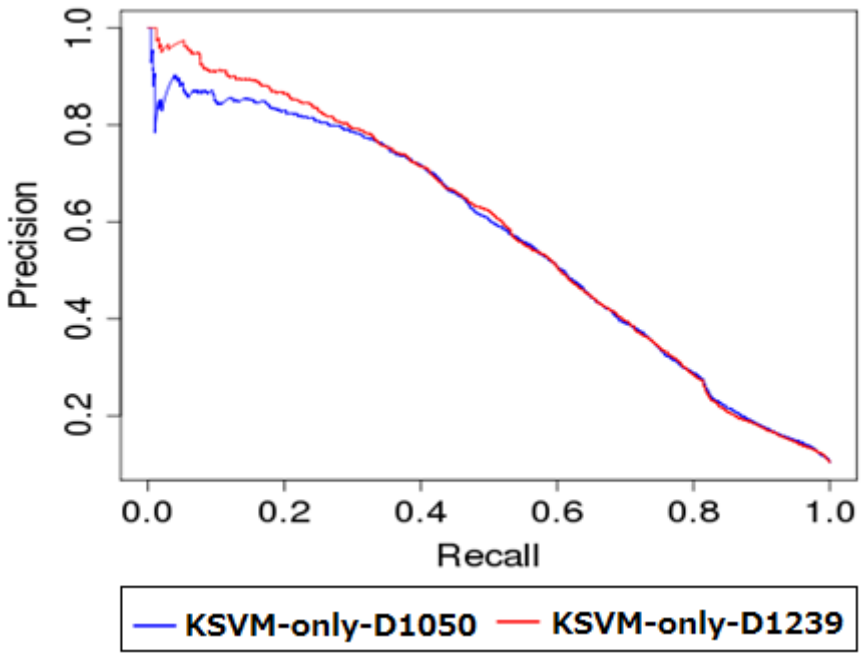


Figure 3.6 PR curves for the datasets with shape string (D1239) and without shape string (D1050) prediction with KSVM as basic classifier

Table 3.3 Performance of KSVM-THR-only, OSD-THR, RUS-THR and RUS-OSD-THR with different decision threshold values on the dataset D1239

Method	**KSVM-THR-only**				**OSD-THR**				**RUS-THR**				**RUS-OSD-THR**			
Thr	**ACC**	**SN**	**SP**	**G**	**ACC**	**SN**	**SP**	**G**	**ACC**	**SN**	**SP**	**G**	**ACC**	**SN**	**SP**	**G**
0.96	89.70	0.14	1.00	3.76	90.66	12.44	99.65	35.21	90.30	20.29	98.35	44.67	91.00	32.91	97.68	56.70
1.73	89.69	0.00	1.00	0.00	89.69	0.00	1.00	0.00	89.67	0.18	99.96	4.20	89.69	0.49	99.94	7.03
8.52	89.69	0.00	1.00	0.00	89.69	0.00	1.00	0.00	89.69	0.00	1.00	0.00	89.69	0.00	1.00	0.00
-2.92	10.30	1.00	0.00	0.00	10.30	1.00	0.00	0.00	10.31	1.00	0.00	0.00	10.31	1.00	0.00	0.00
-1.24	17.85	99.61	8.46	29.03	34.74	98.30	27.44	51.93	19.73	99.26	10.59	32.43	20.99	99.61	11.96	34.51
-0.85	90.11	59.31	93.65	74.53	64.88	92.08	61.75	75.41	37.76	96.36	31.02	54.67	39.13	97.63	32.40	56.24
-0.79	91.60	50.83	96.29	69.96	68.02	90.27	65.46	76.88	41.01	95.40	34.76	57.59	42.02	97.24	35.68	58.90
-0.73	91.99	44.50	97.45	65.85	70.54	89.14	68.40	78.09	43.69	94.56	37.85	59.82	44.65	96.50	38.69	61.10
-0.58	91.97	29.97	99.10	54.50	76.93	84.72	76.04	80.26	51.66	91.34	47.10	65.59	52.28	94.49	47.43	66.95
-0.45	91.57	22.23	99.54	47.04	80.92	80.38	80.98	80.68	57.61	87.45	54.18	68.83	58.00	92.19	54.07	70.60
-0.37	91.32	18.55	99.68	43.01	83.21	77.80	83.83	80.76	61.33	84.66	58.65	70.46	61.50	90.31	58.19	72.49
-0.32	91.17	16.54	99.74	40.62	84.56	75.82	85.57	80.54	63.76	82.40	61.62	71.26	63.95	88.55	61.12	73.57
-0.28	91.07	15.30	99.78	39.08	85.49	74.44	86.76	80.36	65.54	80.88	63.78	71.82	65.72	87.56	63.21	74.39

**Thr = Decision threshold; *ACC = accuracy (%); *SN = sensitivity (%); *SP = specificity (%); *G = G-mean (%)*

Table 3.4 Performance measures comparison of different methods on the dataset D1239

Method	Overall accuracy (%)	Sensitivity (%)	Specificity (%)	G-mean
KSVM-only	90.45	8.02	**99.92**	28.31
OSD	89.61	63.13	92.66	76.48
KSVM-THR-only	**91.07**	15.30	99.78	34.79
OSD-THR	85.49	**74.44**	86.76	**80.36**

Table 3.5 Performance measures comparison on the datasets D1239 and D1050

Data set	Method	Precision (%)	Recall (%)	F-measure (%)
D1050	KSVM-only	**89.18**	4.66	8.86
	OSD	45.27	67.86	**54.31**
	KSVM-THR-only	85.07	11.48	20.24
	OSD-THR	35.84	**77.73**	49.06
D1239	KSVM-only	**92.65**	8.02	14.76
	OSD	49.72	63.13	**55.63**
	KSVM-THR-only	89.09	15.30	26.12
	OSD-THR	39.26	**74.44**	51.40

3.4 Conclusion

In this study, we aimed at the identification of protein-protein interaction sites. The PPI datasets used in this study were highly class-imbalanced, which often decrease classification performance of SVMs. To avoid this issue, we proposed a novel over-sampling technique that effectively utilizes local density of minority samples. We also proposed several methods combined with KSVM-THR and random under-sampling methods to reinforce the tolerance for the class imbalance problem. Experimental results showed that the combination of our OSD algorithm and new feature group led to higher sensitivity, G-mean, precision, MCC, F-measure, and AUC-PR, at least comparable performance with the state-of-the-art methods. In addition, we found that the information of predicted shape strings increase the performance for predicting whether interface or non-interface residues. Further extensions can be considered, for example, combining our algorithm with other heuristic under-sampling method, or feature selection methods.

Chapter 4

Improvement in β-turns Prediction Using Predicted Protein Blocks and Random Under-sampling Method

β-turn is one of the most important reverse turns because of its role in protein folding. Many computational methods techniques for predicting β-turns and their types have been actively studied. However, the performance of prediction is still a challenge. In this study, we utilized predicted protein blocks and position specific scoring matrix together with Random Under-Sampling method to improve the prediction of he β-turns and their types. We performed the experiments and harvested the impressive results on three benchmark datasets that contain 426, 547 and 823 protein sequences, respectively.

4.1 Introduction

Among five types of tight turns, β-turn is one of the most common kinds. β-turn contains four consecutive residues that are not in an α-helix and the distance from the first Cα to the fourth one is less than 7Å [21]. β-turns play an important role in the both conformation and function of protein, such as constructional part of β-hairpins, providing the directional change of the polypeptide [23], and involving in the molecular recognition processes [24]. The formation of β-turn is also a vital step in protein folding [25]. In addition, β-turns make up around 25% number of protein residues.

There are nine β-turn types (I, I', II, II', IV, VIa1, VIa2, VIb and VIII) that are different from the dihedral angles of the two center residues in the turn [26].

Though many researches in beta-turn prediction have been studied [22, 29, 34, 38, 40–42], the performance of methods still be limited. The most recently reported MCC was only 0.5 [42]. In addition, the quality and the number of studies of β-turn types prediction are still low. X.Shi [45] the first time could recognize the rare β-turn types such as I', II' and VI. However, for each turn type, the variance of results on different datasets was high while the distribution was almost similar (Table 4.1).

In this study, we introduce a novel method that can enhance the result of predicting β-turns and their types by using the informative feature groups and dealing with class imbalance problem where the ratio of non-turn residues to the turn residues and the non-specific-type-turn residues to the correct-type-turn residues are high. We present the experimental results on three standard benchmark datasets in comparison with state-of-the-art methods.

4.2 Materials and Methods

4.2.1 Datasets

We utilized three datasets to evaluate the performance of our method. The first one was named BT426 [24] and has been used by many β-turn prediction methods [22, 33, 35–37, 39, 40, 42] as the standard dataset for comparison. The two others were BT547 and BT823, that were used to construct for training and testing COUDES [32]. The numbers of protein sequences in these datasets are 426, 547 and 823, respectively. All

these protein chains contain at least one β-turn and the similarity of each pair chains is less than 25%. Table 4.1 presents the distribution of β-turn types in these datasets. Because of the rare appearance in protein chain, it is hard to predict type VI [32, 33]. Therefore, in this study, just types I, I', II, II', IV and VIII were considered.

The observed turns and their types in protein sequences were assigned by PROMOTIF program [108].

Table 4.1 The type turn's distributions (%) in the datasets

Dataset	I	I'	II	II'	IV	VI	VIII
BT426	9.55	1.29	3.85	0.69	9.48	0.54	2.74
BT547	9.93	1.43	4.05	0.75	9.84	0.62	2.95
BT823	9.87	1.46	3.96	0.77	9.75	0.64	2.70

4.2.2 Feature vector

In this work, PSSMs and predicted Protein Blocks were used as the features for the prediction of β-turns and their types.

Position Specific Scoring Matrices (PSSMs)

The PSSMs were generated by using PSI-BLAST [109] 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. Each element x of these matrices was scaled within the range [0,1] by the logistic function:

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

Predicted Protein Blocks

Predicted secondary structures of protein were effectively applied to predict β-turns and their types [22, 33, 39]. However, the classical classification secondary structure of protein into three states of backbone conformation as α-helix, β-sheet and coil is quite simple, because it lacks the information of the relative orientation of connecting regions. Basing on this kind of classification, 50% total number residues are assigned

as coils while they are believed to belong to a large set of distinct local structures [1, 2].

Many local protein structure libraries that can be able to approximate almost all the local protein structures and do not consider the classical secondary structures were developed to overcome this drawback. These libraries led to the formation of the specific small local structures, named prototypes. A complete set of such prototypes defines a structural alphabet [1].

Protein Blocks (PBs) [3] that can well approximate local protein 3D structures [4] has been successfully applied to many applications at the present time [2, 5]. This structural alphabet consists of sixteen pentapeptide motifs. Each of these prototypes represents a vector of eight average dihedral angles ϕ/ψ, and is labeled as a character in the set of {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p}.

Here, PB-kPRED [110] was used to get the predicted protein blocks. Sixteen characters from A to P symbolized sixteen blocks and X represented the unidentified state. For each residue i in a protein chain, its predicted protein block was represented by a vector of 17 features $(x_i^j)_{17}$, where x_i^j was the probability of residue i as state j.

The feature vector corresponds to each query residue was generated by using a sliding window of size nine amino acids. Thus, there were 333 attributes in one input vector.

4.2.3 Experimental design

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. Support Vector Machines (SVMs) with Gaussian RBF kernel were employed as the basic classifier in this study. Specifically, we used kernlab package (KSVM) [111] to train and test the data.

Since the number of β-turn outnumbers the number of non-turn samples, and for the turn-types prediction problem where the number of each specific type turn samples is many times more than the number of non-specific-type-turn samples, the datasets are imbalanced. This issue results in many positive samples are predicted as negative samples. Many methods have been proposed to handle the class imbalance problem such as over-sampling methods, under-sampling methods, cost-sensitive

methods, and so on [46]. Though SVM is better than these other standard classifiers at dealing with imbalanced data, it often fails when the imbalanced ratio is high [73]. Therefore, in this work, Random Under-Sampling (RUS) was utilized to balance the training datasets before predicting. Grid search relying on MCC to choose the optimal ratio for RUS was operated.

In addition, feature selection based on information gain ratio [96] was applied after under-sampling to reduce the redundant features and achieve the highest MCC.

Figure 4.1 demonstrates the overall architecture of our method.

4.2.4 Filtering

Since a β-turn contains four or more consecutive residues, the output from SVMs needed to be filtered by applying the following rules in order [35]:

i. Change isolated non-turn prediction to turn: tnt → ttt

ii. Change isolated turn prediction to non-turn: ntn → nnn

iii. Change the two non-turn neighbors of two successive turns to turns: nttn → tttt

iv. Change the two non-turn neighbors of three successive turns to turns: ntttn → ttttt

These rules ensure that every final predicted turn is longer than four residues.

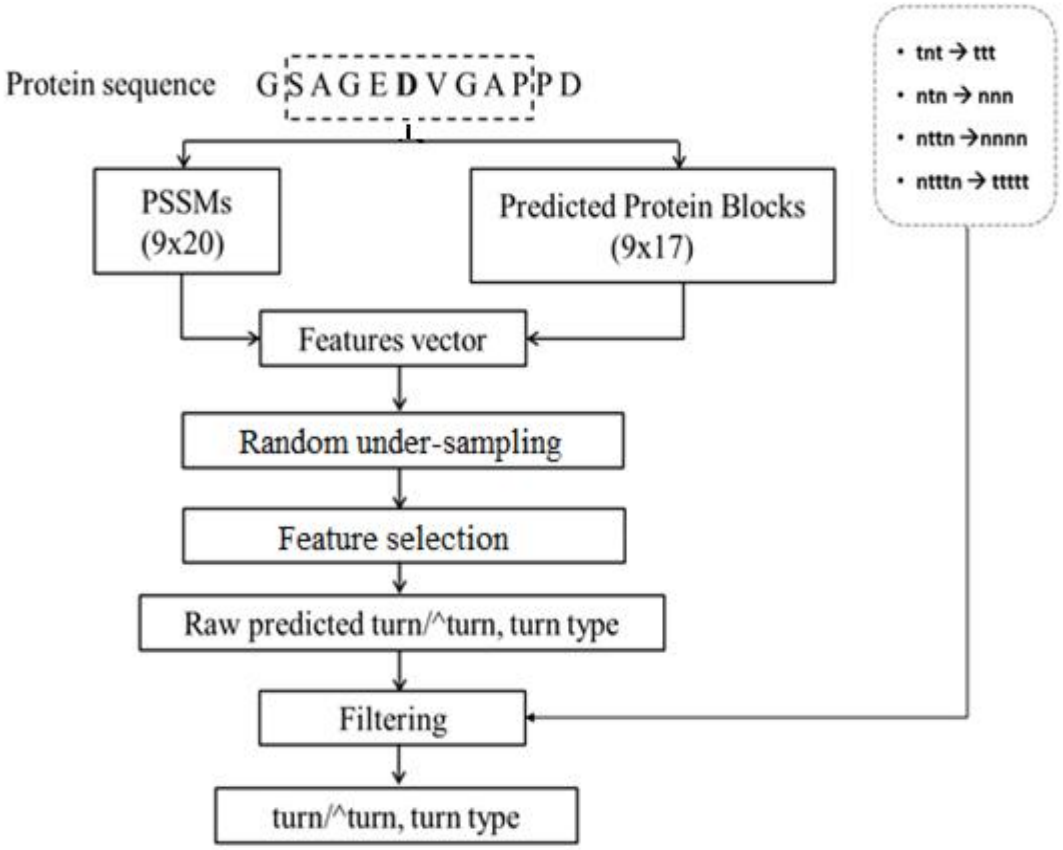


Figure 4.1 The general scheme of our method.

4.2.5 Performance metrics

As MCC, Q_{total}, Q_{obs}, Q_{pred} are often used to measure the quality of β-turn prediction methods [32], they are also used to evaluate the performance of our method and are defined as below:

$$\text{Matthews correlation coefficient (MCC)} = \frac{TP \times TN - FP \times FN}{((TP+FN)\times(TP+FP)\times(TN+FP)\times(TN+FN))^{1/2}}$$

$$Q_{total} = (TP + TN)/(TP + FN + TN + FP)$$

$$Q_{obs} = TP/(TP + FN)$$

$$Q_{pred} = TP/(TP + FP)$$

where TP, TN, FP, FN are the number of true positive, true negative, false positive and false negative samples, respectively. Here, positive sample is the turn or specific type

turn sample; negative sample is the non-turn or non-specific-type-turn sample.

MCC, lies in [-1,1], measures how good the correlation of the predicted and the actual class labels is. It is the most robust measure for β-turn prediction [33].

Besides these metrics, some papers reported specificity value [36, 42] to measure the negative samples prediction ability of predictor, where:

$$\text{Specificity} = \text{TN}/(\text{TN} + \text{FP})$$

In addition, the threshold independent measures ROC (Receiver Operating Characteristics) and AUC (Area Under the Curve), which are often used in bioinformatics [100], are adopted.

4.3 Results and Discussions

4.3.1 Turn/non-turn prediction

The proper choice of sliding window size for extracting the feature vectors affects the performance of prediction. Shepherd [35] showed that window of seven or nine residues was optimal for β-turn prediction. We employed experiments with various sliding window sizes to choose the appropriate one. Table 4.2 presents the results of the sizes from five to eleven residues on the BT426 dataset using PSSMs and predicted protein blocks as features. We selected the size of nine residues since it returns not only the highest MCC but also the highest Q_{total}, Q_{obs} and Q_{pred}.

Table 4.2 The evaluation results of using different window sizes for PSSM values and predicted protein blocks without under-sampling and feature selection on the BT426 dataset

Window size	Q_{total}(%)	Q_{obs}(%)	Q_{pred}(%)	MCC
5	84.7	56.7	74.0	0.55
7	85.0	58.2	74.6	0.56
9	**85.2**	**58.6**	**75.1**	**0.57**
11	84.9	57.8	74.6	0.56

Experiments to value the impact of evolutionary information PSSMs, predicted protein block, and their combination were also performed. Table 4.3 presents the effect of each kind of features on β-turn prediction. The much higher MCC, Q_{total}, Q_{obs}

and Q_{pred} in the two cases of using predicted protein blocks in comparison of using PSSMs on the three datasets demonstrates the importance of this group of features. Figure 4.2 shows that predicted protein blocks is more effective than PSSMs in the area of true positive rate lower than 0.9 for the dataset BT426, and 0.85 for BT547 and BT823 datasets; and the combination of these two feature groups produces the best result on all three datasets.

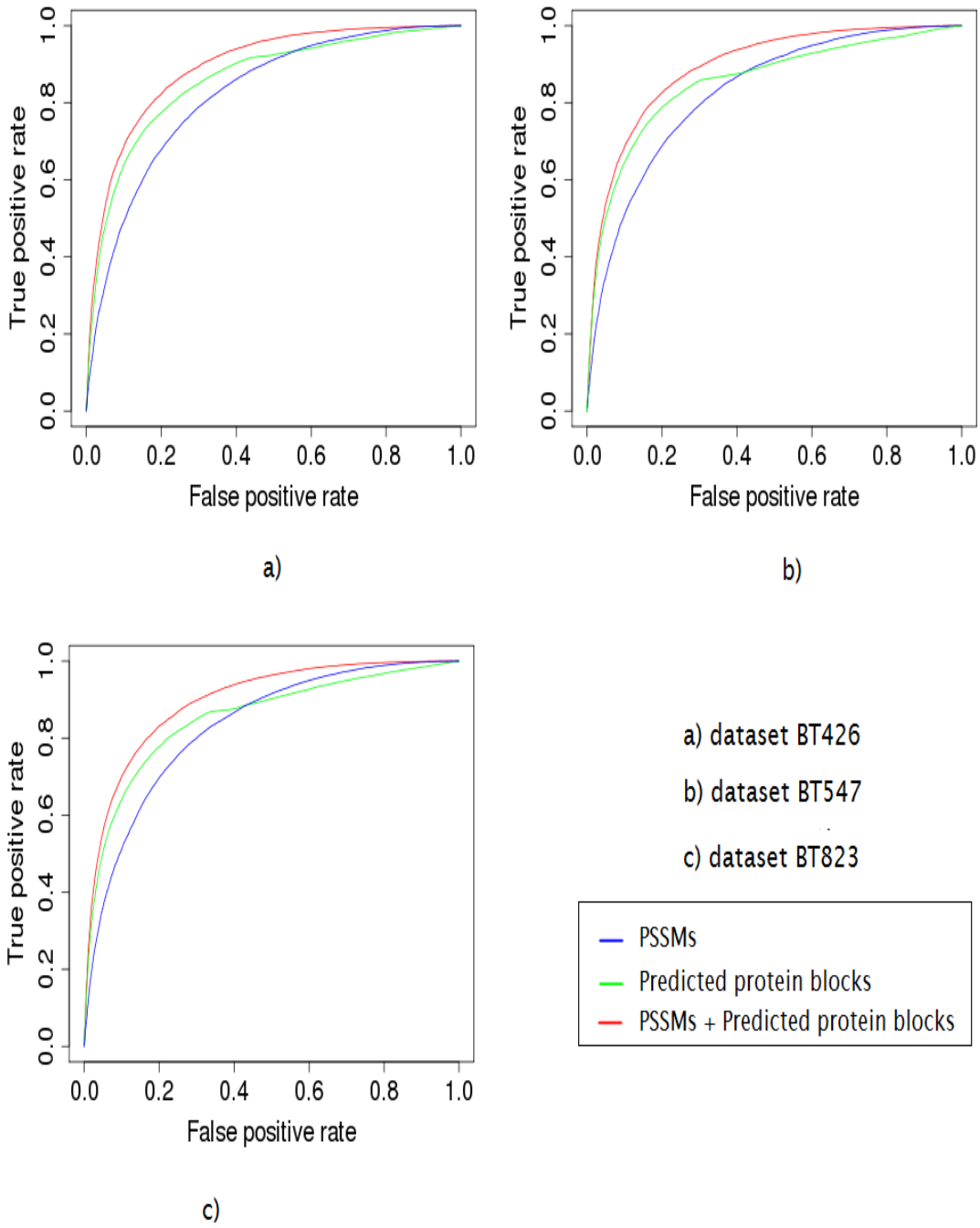


Figure 4.2 ROC curves for the comparison of various feature groups, without feature selection on the BT426, BT547 and BT823 datasets.

Table 4.3 The evaluation results of the three datasets using different kinds of feature groups with sliding window size of 9, without under-sampling and feature selection

Dataset	Feature group	Q_{total} (%)	Q_{obs} (%)	Q_{pred} (%)	MCC
BT426	PSSMs	79.90	33.87	67.17	0.37
	Predicted protein blocks	83.76	52.70	73.00	0.52
	PSSMs + Predicted protein blocks	**85.24**	**58.60**	**75.19**	**0.57**
BT547	PSSMs	79.92	38.38	69.15	0.40
	Predicted protein blocks	83.66	54.43	74.66	0.54
	PSSMs + Predicted protein blocks	**84.72**	**59.68**	**75.31**	**0.57**
BT823	PSSMs	80.39	37.55	70.00	0.40
	Predicted protein blocks	83.97	53.05	75.55	0.53
	PSSMs + Predicted protein blocks	**85.38**	**59.31**	**76.90**	**0.58**

The comparison of our method with the other competitive methods on the BT426 dataset is presented in Table 4.4. It shows that our method outperformed KLR and the others with MCC of 0.585.

Table 4.3 and Table 4.4 show that the use of under-sampling and feature selection for eliminating negative samples and redundant features to relax the class-imbalance, not only increased Q_{obs} (12.41%) but also MCC (0.015). Figure 4.3 displays the ROC curves of our method and KLR that was taken from [42].

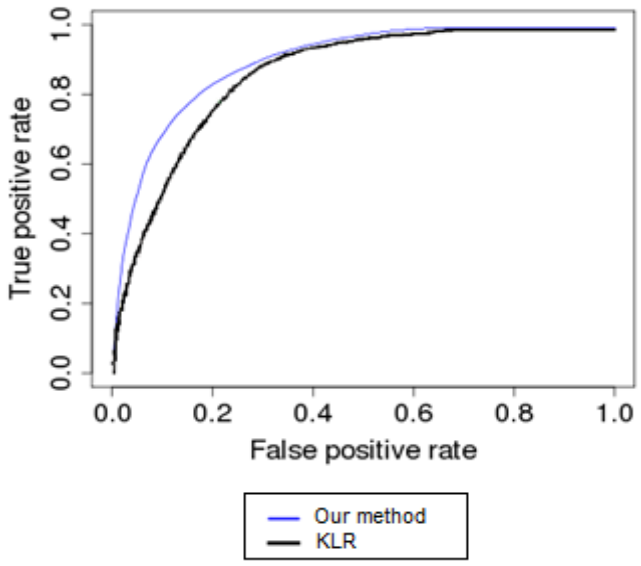


Figure 4.3 ROC curves of KLR and our method on the BT426 dataset.

Table 4.5 presents the results of the competing methods on the datasets BT547 and BT823, with our method achieved the highest values on MCC, Q_{total} and Q_{pred}. ROC curves of our methods on these two datasets are shown in Figure 4.4.

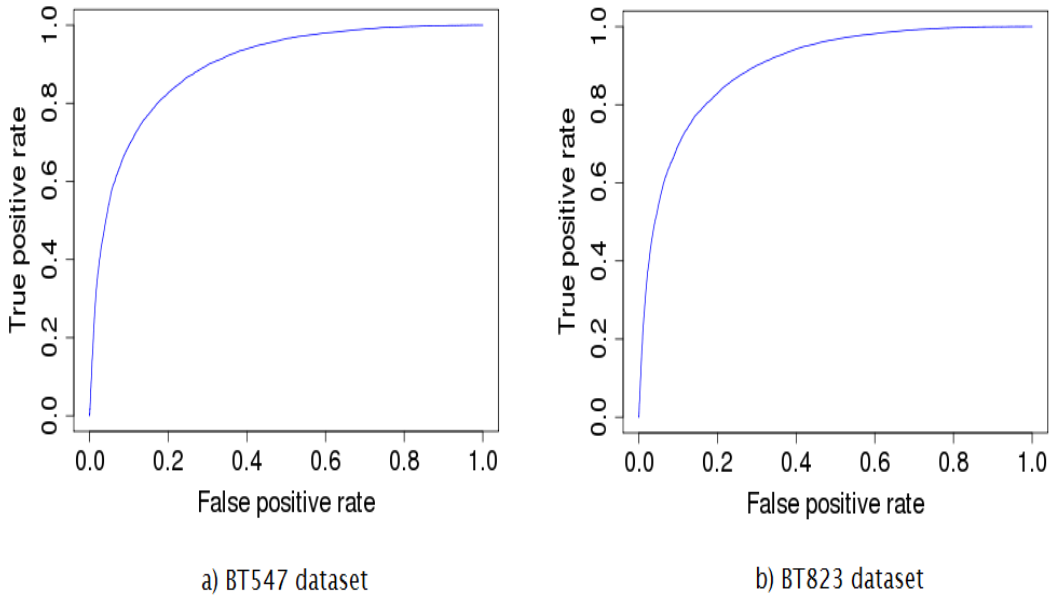


Figure 4.4 ROC curves on BT547 and BT823 datasets.

Table 4.4 Comparison of competitive methods on the BT426 dataset. "_" means this value was not reported

Method	Q_{total} (%)	Q_{obs} (%)	Q_{pred} (%)	Specificity (%)	MCC	AUC
Our method	**84.41**	71.01	**66.89**	**88.71**	**0.585**	**0.893**
KLR [42]	80.4	65.25	58.98	85.34	0.50	0.86
NetTurnP [36]	78.2	**75.6**	54.4	79.1	0.50	0.86
DEBT [33]	79.2	70.1	54.8	-	0.48	0.84
BTNpred [40]	80.9	55.6	62.7	-	0.47	-
SVM [39]	79.8	68.9	55.6	-	0.47	0.87
BTSVM [37]	78.7	62.0	56.0	-	0.45	-
BetaTPred [22]	75.5	72.3	49.8	-	0.43	-
BTPRED [35]	74.9	48.0	55.3	-	0.35	-

4.3.2 Turn types prediction

Our performance of β-turn types prediction on the three datasets BT426, BT547, BT823 is shown in Table 4.6. All the AUC values are higher than 0.7, and most of them are higher than 0.85. It proofs that our method is acceptable in predicting β-turn type [42].

Table 4.7 presents the MCC of competing methods. While DEBT cannot predict type I' and II', our methods achieved the highest MCC in comparison with the other method on all three datasets (0.635 and 0.530 on BT426; 0.632 and 0.453 on BT547; 0.635 and 0.454 on BT823 for type I' and II', respectively). Though MCC of X.Shi et al. was higher than our in some cases, our method appeared to be stable on the three datasets. For example, MCC of X.Shi et al. on type VIII of dataset BT426 decreased from 0.246 to 0.044 on dataset BT547, or from 0.714 to 0.529 on type I. It shows that the performance of this method was quite dependent on the specific dataset. ROC curves of our β-turn types predictions are shown in Figure 4.5.

Table 4.5 Comparison of competitive methods on the BT547 and BT823 datasets. "_" means this value was not reported

Dataset	Method	Q_{total} (%)	Q_{obs} (%)	Q_{pred} (%)	Specificity (%)	MCC	AUC
BT547	Our method	**85.01**	64.70	**73.37**	**91.96**	**0.591**	**0.894**
	KLR [42]	80.46	65.36	59.04	-	0.50	-
	DEBT [33]	80.0	68.7	55.9	-	0.49	0.85
	BTNpred [40]	80.5	54.2	61.6	-	0.45	-
	SVM [39]	76.6	70.2	47.6	-	0.43	-
	COUDES [32]	74.6	**70.4**	48.7	-	0.42	-
BT823	Our method	**84.96**	68.46	**70.51**	**90.46**	**0.595**	**0.896**
	KLR [42]	80.66	64.64	58.42	-	0.49	-
	DEBT [33]	80.9	66.1	55.9	-	0.48	0.84
	BTNpred [40]	80.6	54.6	60.8	-	0.45	-
	SVM [39]	76.8	**72.3**	53.0	-	0.45	-
	COUDES [32]	74.2	69.6	47.5	-	0.41	-

Table 4.6 Beta-turn types predicting results of our method on the BT426, BT547 and BT823 datasets

Dataset	β-turn type	Q_{total} (%)	Q_{obs} (%)	Q_{pred} (%)	Specificity (%)	MCC	AUC
BT426	I	91.65	64.30	55.45	94.54	0.551	0.915
	I’	99.11	60.83	67.36	99.61	0.635	0.968
	II	94.88	81.59	41.64	95.42	0.561	0.963
	II’	99.35	53.26	53.44	99.67	0.530	0.977
	IV	78.72	66.18	25.78	80.03	0.315	0.823
	VIII	82.91	69.45	10.51	83.29	0.223	0.847
BT547	I	91.21	64.21	54.93	94.18	0.545	0.916
	I’	99.00	60.45	67.16	99.57	0.632	0.972
	II	96.03	70.40	50.83	97.12	0.578	0.965
	II’	99.35	32.70	63.58	99.85	0.453	0.942
	IV	78.79	66.30	26.74	80.16	0.322	0.825
	VIII	85.32	64.43	12.26	85.95	0.235	0.859
BT823	I	91.63	63.53	56.82	94.71	0.554	0.917
	I’	98.99	60.50	67.84	99.57	0.635	0.974
	II	96.40	68.30	53.68	97.56	0.587	0.964
	II’	99.31	35.54	59.02	99.80	0.454	0.952
	IV	78.46	68.08	26.49	79.59	0.326	0.827
	VIII	86.69	60.57	11.82	87.42	0.225	0.861

Table 4.7 MCCs comparison between the competitive methods. “_” means this value was not reported

Dataset	Method	I	I’	II	II’	IV	VIII
BT426	Our method	0.551	**0.635**	0.561	**0.530**	0.315	0.223
	X.Shi et al. [45]	**0.714**	0.513	**0.684**	0.415	**0.459**	**0.246**
	NetTurnP[36]	0.36	0.23	0.31	0.16	0.27	0.16
	DEBT[33]	0.36	_	0.29	_	0.27	0.14
	COUDES [32]	0.309	0.226	0.302	0.106	0.109	0.071
BT547	Our method	**0.545**	**0.632**	**0.578**	**0.453**	**0.322**	**0.235**
	X.Shi et al. [45]	0.529	0.538	0.548	0.337	0.311	0.044
	DEBT[33]	0.38	_	0.33	_	0.27	0.14
BT823	Our method	0.554	**0.635**	0.587	**0.454**	**0.326**	**0.225**
	X.Shi et al. [45]	**0.636**	0.416	**0.630**	0.361	0.317	0.125
	DEBT[33]	0.39	_	0.33	_	0.27	0.14

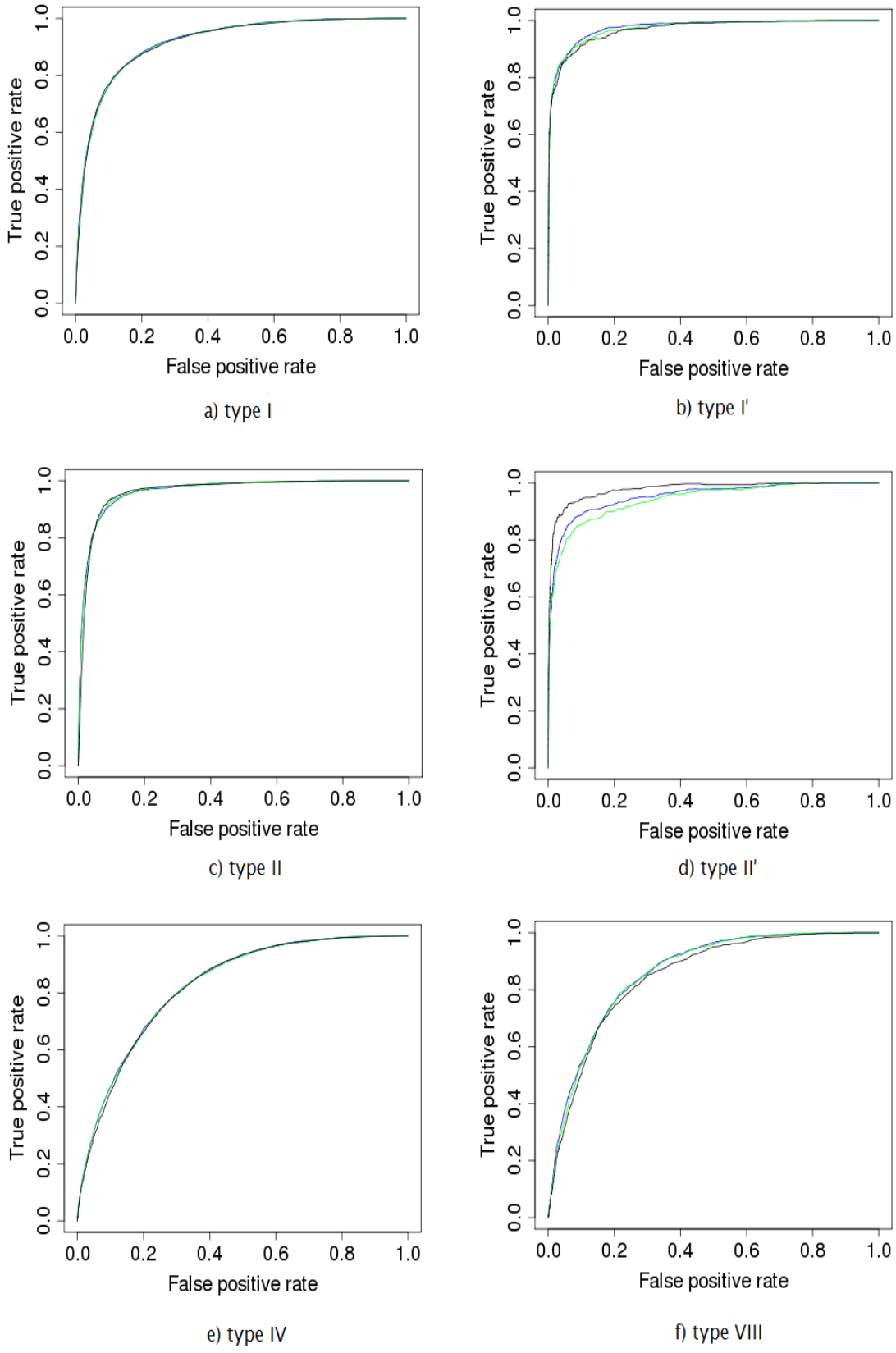


Figure 4.5 ROC curves of our method on the three datasets BT426 (black), BT547 (green), and BT823 (blue).

4.4 Conclusions

In this study, we presented a new method to identify the β-turns and their types in protein sequence. We focused on both using more the well-characterized features and class-imbalanced-dealt technique. We achieved the highest MCCs of 0.585, 0.591 and 0.595 on the three datasets BT426, BT547 and BT823, respectively, in comparison with the state-of-the-art β-turns prediction methods. In the field of β-turn types prediction, we also harvested the high and stable results. Further extension can be considered such as using the effective method to handle the class-imbalanced problem.

Chapter 5

Conclusions

The previous chapters introduced the problems, proposed the methods to improve the performance of predicting protein-protein interaction site and β-turn. This chapter summarizes our works, and suggests some ideas for the future works.

5.1 Dissertation Summary

Proteins are very important because they are involved in many functions in a living cell. Most proteins perform their functions via protein-protein interactions to maintain the organism's life. However, many interactions between proteins are unidentified until now. Therefore, study the mechanism of protein-protein interactions, especially, which part in protein sequence has the contacted ability, is one of the necessary problems in bioinformatics.

Nevertheless, to clearly understand the protein-protein interaction sites as well as the other functions of proteins, it is necessary to understand their three-dimensional structure. One of the most important tasks in this field is learning about β-turns and their types.

In this thesis, we aimed at (i) improving the performance of protein-protein interaction sites prediction using a novel over-sampling method and informative

features; and (ii) improving the prediction of β-turns and their types by applying predicted protein blocks and under-sampling techniques. The main contributions of our thesis are listed below.

Firstly, the datasets we used for protein-protein interaction sites prediction were highly class-imbalanced. Thus, when using SVMs for prediction, the performance often fails. To overcome this drawback, we proposed a new method that over-sampled the training set before classifying, and it was effective in this case. The combinations of our new algorithm with KSVM-THR and random under-sampling methods were also proposed. Experimental results showed that our new methods achieved higher sensitivity, precision, G-mean, F-measure, and AUC-PR than the state-of-the-art methods. We also found that the predicted shape strings were informative for predicting whether interface or non-interface residues.

Secondly, we investigated the information of predicted protein blocks and applied for β-turns prediction. The use of this feature can improve the performance of prediction, in comparison with the most recent publication. Once again, resampling strategy was used to deal with the class imbalance. Specifically, in this study, we utilized random under-sampling method. In addition, feature selection based on gain information ratio was applied to remove redundant features. We also performed the β-turn types prediction to recognize which type of turn that residue belonged to. Results of experiments on three standard benchmark datasets showed that our methods are comparable with the state -of-the-art methods.

5.2 Future Works

The methods to deal with imbalanced datasets are very important because the class imbalance problems exist everywhere in the real world, especially in the realm of biological datasets. In this thesis, we developed the new algorithm OSD to over-sample the minority set of an imbalanced dataset by focusing on the local density. This algorithm was applied to improve the prediction of protein-protein interaction sites. Though we achieved good results, further extensions can be considered.

Firstly, OSD just handles the numerical values but the nominal values. Thus, the extension of OSD can be thought about so that it can be applied for the datasets with nominal features. Secondly, because feature selection affects the performance of

prediction on imbalanced dataset, we can combine feature selection with our methods, as a preprocessing step. It may lead to improve the results. In addition, random under-sampling is the most naïve under-sampling method. This method is simple and fast, however, leads to lose many informations. Our experiment showed that reducing the number of majority samples before applying the other methods could create the good model. Thus, the use of better under-sampling method may result in better performance than random under-sampling.

About the second problem in our thesis, the β-turn prediction, we also think about applying the under-sampling technique that is better than random under-sampling. Since the model that was created by utilizing PSSMs, predicted protein block, under-sampling and feature selection returns good results in this situation, it also can be used for predicting protein-protein interactions sites and the other kinds of tight turn such as α-turn or γ-turn.

In addition, in this study, residues belong to β-turn type VI were not predicted because of the limitation of their appearances in a protein chain. However, recognizing these residues is as important as identifying the other kinds of residue in the sequence. Thus, we aim to develop our method that in the future, we can recognize all the β-turn types.

Bibliography

1. Offmann B, Tyagi M, De Brevern AG: **Local Protein Structures**. *Current Bioinformatics* 2007, **2**:38.

2. Joseph AP, Agarwal G, Mahajan S, Gelly J-C, Swapna LS, Offmann B, Cadet F, Bornot A, Tyagi M, Valadié H, Schneider B, Etchebest C, Srinivasan N, De Brevern AG: **A short survey on protein blocks**. *Biophysical Reviews* 2010, **2**:137–145.

3. De Brevern AG, Etchebest C, Hazout S: **Bayesian probabilistic approach for predicting backbone structures in terms of protein blocks.** *Proteins* 2000, **41**:271–87.

4. De Brevern AG: **New Assessment of a Structural Alphabet**. *In Silico Biology* 2005, **5**:283–289.

5. Joseph AP, Srinivasan N, De Brevern AG: **Improvement of protein structure comparison using a structural alphabet**. *Biochimie* 2011, **93**:1434–45.

6. *Bioinformatics: A Concept-Based Introduction*. Boston, MA: Springer US; 2009.

7. Keskin O, Tuncbag N, Gursoy A: **Characterization and prediction of protein interfaces to infer protein-protein interaction networks.** *Current pharmaceutical biotechnology* 2008, **9**:67–76.

8. Wang B, Chen P, Huang D-S, Li J, Lok T-M, Lyu MR: **Predicting protein interaction sites from residue spatial sequence profile and evolution rate**. *FEBS Letters* 2006, **580**:380–4.

9. Browne F, Zheng H, Wang H, Azuaje F: **From Experimental Approaches to Computational Techniques: A Review on the Prediction of Protein-Protein Interactions**. *Advances in Artificial Intelligence* 2010, **2010**:1–15.

10. Wells JA: **[18] Systematic mutational analyses of protein-protein interfaces**. *Methods in Enzymology* 1991, **202**:390–411.

11. Ezkurdia I, Bartoli L, Fariselli P, Casadio R, Valencia A, Tress ML: **Progress and challenges in predicting protein-protein interaction sites**. *Briefings in Bioinformatics* 2009, **10**:233–46.

12. Fernández-Recio J: **Prediction of protein binding sites and hot spots**. *WIREs Comput Mol Sci* 2011, **1**:680–698.

13. Li N, Sun Z, Jiang F: **Prediction of protein-protein binding site by using core interface residue and support vector machine**. *BMC Bioinformatics* 2008, **9**:553.

14. Li M-H, Lin L, Wang X-L, Liu T: **Protein-protein interaction site prediction based on conditional random fields.** *Bioinformatics (Oxford, England)* 2007, **23**:597–604.

15. Fariselli P, Pazos F, Valencia A, Casadio R: **Prediction of protein-protein interaction sites in heterocomplexes with neural networks.** *European Journal of Biochemistry* 2002, **269**:1356–61.

16. Chen X, Jeong JC: **Sequence-based prediction of protein interaction sites with an integrative method.** *Bioinformatics (Oxford, England)* 2009, **25**:585–91.

17. Kini RM, Evans HJ: **Prediction of potential protein-protein interaction sites from amino acid sequence: Identification of a fibrin polymerization site**. *FEBS Letters* 1996, **385**:81–6.

18. Chen P, Li J: **Sequence-based identification of interface residues by an integrative profile combining hydrophobic and evolutionary information**. *BMC Bioinformatics* 2010, **11**:402.

19. Ofran Y, Rost B: **ISIS: interaction sites identified from sequence.** *Bioinformatics (Oxford, England)* 2007, **23**:e13–6.

20. Res I, Mihalek I, Lichtarge O: **An evolution based classifier for prediction of protein interfaces without using protein structures**. *Bioinformatics (Oxford, England)* 2005, **21**:2496–501.

21. Chou K-C: **Prediction of Tight Turns and Their Types in Proteins**. *Analytical Biochemistry* 2000, **286**:1–16.

22. Kaur H, Raghava GPS: **Prediction of beta-turns in proteins from multiple alignment using neural network**. *Protein Science* 2003, **12**:627–634.

23. Marcelino AMC, Gierasch LM: **Roles of beta-turns in protein folding: from peptide models to protein engineering.** *Biopolymers* 2008, **89**:380–91.

24. Guruprasad K, Rajkumar S: **Beta-and gamma-turns in proteins revisited: a new set of amino acid turn-type dependent positional preferences and potentials**. *Journal of Biosciences* 2000, **25**:143–56.

25. Takano K, Yamagata Y, Yutani K: **Role of amino acid residues at turns in the conformational stability and folding of human lysozyme**. *Biochemistry* 2000, **39**:8655–65.

26. Hutchinson EG, Thornton JM: **A revised set of potentials for beta-turn formation in proteins**. *Protein Science* 1994, **3**:2207–2216.

27. Chou PY, Fasman GD: **Conformational parameters for amino acids in helical, β-sheet, and random coil regions calculated from proteins**. *Biochemistry* 1974, **13**:211–222.

28. Wilmot CM, Thornton JM: **Analysis and prediction of the different types of beta-turn in proteins**. *Journal of Molecular Biology* 1988, **203**:221–32.

29. Wilmot CM, Thornton JM: **Beta-turns and their distortions: a proposed new nomenclature.** *Protein Engineering* 1990, **3**:479–93.

30. Chou KC, Blinn JR: **Classification and prediction of beta-turn types**. *Journal of Protein Chemistry* 1997, **16**:575–95.

31. Zhang C-T, Chou K-C: **Prediction of β-turns in proteins by 1-4 and 2-3 correlation model**. *Biopolymers* 1997, **41**:673–702.

32. Fuchs PFJ, Alix AJP: **High accuracy prediction of beta-turns and their types using propensities and multiple alignments.** *Proteins* 2005, **59**:828–39.

33. Kountouris P, Hirst JD: **Predicting beta-turns and their types using predicted backbone dihedral angles and secondary structures**. *BMC Bioinformatics* 2010, **11**:407.

34. McGregor MJ, Flores TP, Sternberg MJ: **Prediction of beta-turns in proteins using neural networks.** *Protein Engineering* 1989, **2**:521–6.

35. Shepherd AJ, Gorse D, Thornton JM: **Prediction of the location and type of beta-turns in proteins using neural networks**. *Protein Science* 1999, **8**:1045–1055.

36. Petersen B, Lundegaard C, Petersen TN: **NetTurnP – Neural Network Prediction of Beta-turns by Use of Evolutionary Information and Predicted Protein Sequence Features**. *PloS ONE* 2010, **5**:e15079.

37. Pham TH, Satou K, Ho TB: **Prediction and analysis of beta-turns in proteins by support vector machine**. *Genome Informatics* 2003, **14**:196–205.

38. Zhang Q, Yoon S, Welsh WJ: **Improved method for predicting beta-turn using support vector machine.** *Bioinformatics (Oxford, England)* 2005, **21**:2370–4.

39. Hu X, Li Q: **Using support vector machine to predict beta- and gamma-turns in proteins**. *Journal of Computational Chemistry* 2008, **29**:1867–75.

40. Zheng C, Kurgan L: **Prediction of beta-turns at over 80% accuracy based on an ensemble of predicted secondary structures and multiple alignments**. *BMC Bioinformatics* 2008, **9**:430.

41. Cai Y-D, Liu X-J, Li Y-X, Xu X, Chou K-C: **Prediction of beta-turns with learning machines.** *Peptides* 2003, **24**:665–9.

42. Elbashir MK, Wang J, Wu F, Li M: **Sparse Kernel Logistic Regression for β -turns Prediction**. *Systems Biology (ISB), 2012 IEEE 6th International Conference on* 2012:246–251.

43. Kaur H, Raghava GPS: **A neural network method for prediction of beta-turn types in proteins using evolutionary information**. *Bioinformatics (Oxford, England)* 2004, **20**:2751–8.

44. Kirschner A, Frishman D: **Prediction of beta-turns and beta-turn types by a novel bidirectional Elman-type recurrent neural network with multiple output layers (MOLEBRNN).** *Gene* 2008, **422**:22–9.

45. Shi X, Hu X, Li S, Liu X: **Prediction of β-turn types in protein by using composite vector**. *Journal of Theoretical Biology* 2011, **286**:24–30.

46. He H, Garcia EA: **Learning from Imbalanced Data**. *IEEE Transactions on Knowledge and Data Engineering* 2009, **21**:1263–1284.

47. Barandela R, Sánchez J, García V, Rangel E: **Strategies for learning in class imbalance problems**. *Pattern Recognition* 2003, **36**:849–851.

48. Sun Y, Wong AKC, Kamel MS: **Classification of Imbalanced Data: a Review**. *International Journal of Pattern Recognition and Artificial Intelligence* 2009, **23**:687–719.

49. Kotsiantis S, Kanellopoulos D, Pintelas P: **Handling imbalanced datasets: A review**. *International Transactions on Computer Science and Engineering* 2006, **30**:25–36.

50. Mani I, Zhang J: **kNN approach to unbalanced data distributions: a case study involving information extraction**. In *Proceedings of Workshop on Learning from Imbalanced Datasets*. 2003.

51. Phua C, Alahakoon D, Lee V: **Minority report in fraud detection**. *ACM SIGKDD Explorations Newsletter* 2004, **6**:50.

52. Chan PK, Fan W, Prodromidis AL, Stolfo SJ: **Distributed data mining in credit card fraud detection**. *IEEE Intelligent Systems* 1999, **14**:67–74.

53. Kubat M, Holte RC, Matwin S: **Machine Learning for the Detection of Oil Spills in Satellite Radar Images**. *Machine Learning* 1998, **30**:195–215.

54. Kazuo Ezawa MS: **Learning Goal Oriented Bayesian Networks for Telecommunications Risk Management**. In *Proceedings of the 13th International Conference on Machine Learning*. Morgan Kaufmann; 1996:139–147.

55. Cardie C: **Improving minority class prediction using case-specific feature weights**. In *Proceedings of the Fourteenth International Conference on Machine Learning*. Morgan Kaufmann; 1997:57–65.

56. Yousef M, Nebozhyn M, Shatkay H, Kanterakis S, Showe LC, Showe MK: **Combining multi-species genomic data for microRNA identification using a Naive Bayes classifier**. *Bioinformatics (Oxford, England)* 2006, **22**:1325–34.

57. Ofran Y, Rost B: **Predicted protein-protein interaction sites from local sequence information**. *FEBS Letters* 2003, **544**:236–9.

58. Sikić M, Tomić S, Vlahovicek K: **Prediction of protein-protein interaction sites in sequences and 3D structures by random forests.** *PLoS Computational Biology* 2009, **5**:e1000278.

59. Yu D-J, Hu J, Tang Z-M, Shen H-B, Yang J, Yang J-Y: **Improving protein-ATP binding residues prediction by boosting SVMs with random under-sampling**. *Neurocomputing* 2013, **104**:180–190.

60. Batuwita R, Palade V: **microPred: effective classification of pre-miRNAs for human miRNA gene prediction.** *Bioinformatics (Oxford, England)* 2009, **25**:989–995.

61. Anand A, Pugalenthi G, Fogel GB, Suganthan PN: **An approach for classification of highly imbalanced data using weighting and undersampling**. *Amino Acids* 2010, **39**:1385–1391.

62. Han K: **Effective sample selection for classification of pre-miRNAs.** *Genetics and Molecular Research : GMR* 2011, **10**:506–18.

63. García-Pedrajas N, Pérez-Rodríguez J, García-Pedrajas M, Ortiz-Boyer D, Fyfe C: **Class imbalance methods for translation initiation site recognition in DNA sequences**. *Knowledge-Based Systems* 2012, **25**:22–34.

64. Visa S: **Issues in Mining Imbalanced Data Sets - A Review Paper**. In *Proceedings of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference*. 2005:67–73.

65. Cover T, Hart P: **Nearest neighbor pattern classification**. *IEEE Transactions on Information Theory* 1967, **13**:21–27.

66. Quinlan JR: **Induction of Decision Trees**. *Machine Learning* 1986, **1**:81–106.

67. Quinlan JR: *C4.5: programs for machine learning*. Morgan Kaufmann; 1993.

68. Carvajal K, Chacon M, Mery D, Acuna G: **Neural network method for failure detection with skewed class distribution**. *Insight* , **46**:399–402.

69. Vapnik V, Lerner A: **Pattern Recognition using Generalized Portrait Method**. *Automation and Remote Control* 1963, **24**.

70. Japkowicz N, Stephen S: **The class imbalance problem: A systematic study**. *Intelligent Data Analysis* 2002, **6**:429–449.

71. Veropoulos K, Campbell C, Cristianini N: **Controlling the Sensitivity of Support Vector Machines**. In *Proceedings of the International Joint Conference on AI*. 1999:55–60.

72. Wu G, Chang E: **Class-Boundary Alignment for Imbalanced Dataset Learning**. In *ICML 2003 Workshop on Learning from Imbalanced Data Sets*. 2003:49–56.

73. Akbani R, Kwek S, Japkowicz N: **Applying support vector machines to imbalanced datasets**. In *Proceedings of the 15th European Conference on Machine Learning*. 2004:39–50.

74. Ganganwar V: **An overview of classification algorithms for imbalanced datasets**. *International Journal of Emerging Technology and Advanced Engineering* 2012, **2**:42–47.

75. Chawla N V., Bowyer KW, Hall LO, Kegelmeyer WP: **SMOTE : Synthetic Minority Over-sampling Technique**. *Journal of Artificial Intelligence Research* 2002, **16**:321–357.

76. Blagus R, Lusa L: **SMOTE for high-dimensional class-imbalanced data**. *BMC Bioinformatics* 2013, **14**:106.

77. Chawla N V, Lazarevic A, Hall LO, Bowyer K: **SMOTEBoost : Improving Prediction of the Minority Class in Boosting**. In *Proceedings of the Principles of Knowledge Discovery in Databases, PKDD-2003*. 2003:107–119.

78. Ramentol E, Caballero Y, Bello R, Herrera F: **SMOTE-RSB *: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory**. *Knowledge and Information Systems* 2011, **33**:245–265.

79. Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C: **Safe-Level-SMOTE : Safe-Level-Synthetic Minority Over-Sampling TEchnique**. In *Advances in Knowledge Discovery and Data Mining*. Springer Berlin Heidelberg; 2009:475–482.

80. Hui Han,Wenyuan Wang BM, Han H, Wang W, Mao B: **Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning**. In *Advances in Intelligent Computing*. 2005:878 – 887.

81. Jo T, Japkowicz N: **Class Imbalances versus Small Disjuncts**. *ACM SIGKDD Explorations Newsletter* 2004, **6**:40–49.

82. Liu X, Wu J, Zhou Z: **Exploratory Undersampling for Class-Imbalance Learning**. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 2009, **39**:539–550.

83. Zadrozny B, Elkan C: **Learning and making decisions when costs and probabilities are both unknown**. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining - KDD'01*. New York: ACM Press; 2001:204–213.

84. Quinlan JR: **Improved Estimates for the Accuracy of Small Disjuncts**. *Machine Learning* 1991, **6**:93–98.

85. Du S, Chen S: **Weighted support vector machine for classification**. *2005 IEEE International Conference on Systems, Man and Cybernetics* , **4**:3866–3871.

86. Yang X, Song Q, Cao A: **Weighted support vector machine for data classification**. In *Proceedings of the International Joint Conference on Neural Networks*. Montreal: IEEE; 2005, **2**:859–864.

87. Elkan C: **The foundations of cost-sensitive learning**. In *Proceeding IJCAI'01 Proceedings of the 17th international joint conference on Artificial intelligence*. Morgan Kaufmann Publishers Inc.; 2001:973–978.

88. Ting KM: **An instance-weighting method to induce cost-sensitive trees**. *IEEE Transactions on Knowledge and Data Engineering* 2002, **14**:659–665.

89. Chen JJ, Tsai C-A, Moon H, Ahn H, Young JJ, Chen C-H: **Decision threshold adjustment in class prediction**. *SAR and QSAR in environmental researchnvironmental Research* 2006, **17**:337–52.

90. Lin W-J, Chen JJ: **Class-imbalanced classifiers for high-dimensional data**. *Briefings in Bioinformatics* 2013, **14**:13–26.

91. Cohen WW: **Fast Effective Rule Induction**. In *Proceedings of the Twelfth International Conference on Machine Learning*. Morgan Kaufmann; 1995:115–123.

92. Juszczak P, Duin RPW: **Uncertainty sampling methods for one-class classifiers**. In *Proceedings of the ICML'03 Workshop on Learning from Imbalanced Data Sets*. 2003:5.

93. Raskutti B, Kowalczyk A: **Extreme re-balancing for SVMs: a case study**. *ACM SIGKDD Explorations Newsletter* 2004, **6**:60.

94. Saeys Y, Inza I, Larrañaga P: **A review of feature selection techniques in bioinformatics**. *Bioinformatics (Oxford, England)* 2007, **23**:2507–17.

95. Van Der Putten P, Van Someren M: **A Bias-Variance Analysis of a Real World Learning Problem: The CoIL Challenge 2000**. *Machine Learning* 2004, **57**:177–195.

96. Altidor W, Khoshgoftaar TM, Hulse J Van: **Robustness of Filter-Based Feature Ranking: A Case Study.** In *Proceedings of 24th Florida Arti cial Intelligence Research Society Conference (FLAIRS-24)*. Palm Beach, FL: 2011:453–458.

97. Veeraswamy A, Balamurugan DSAA: **A Survey of Feature Selection Algorithms in Data Mining**. *International Journal of Advanced Research In Technology* 2011, **1**:108–117.

98. Kohavi R, John GH: **Wrappers for Feature Subset Selection**. *Artificial Intelligence* 1997, **97**:273 – 324.

99. Joshi M V., Kumar V, Agarwal RC: **Evaluating Boosting Algorithms to Classify Rare Classes: Comparison and Improvements**. In *Proceedings IEEE International Conference on Data Mining*. IEEE Computer Society; 2001:257–264.

100. Sonego P, Kocsor A, Pongor S: **ROC analysis: applications to the classification of biological sequences and 3D structures**. *Briefings in Bioinformatics* 2008, **9**:198–209.

101. Fawcett T: **An introduction to ROC analysis**. *Pattern Recognition Letters* 2006, **27**:861–874.

102. Wang D, Li T, Sun J, Li D, Xiong W, Wang W, Tang S: **Shape string: a new feature for prediction of DNA-binding residues**. *Biochimie* 2013, **95**:354–358.

103. Zhu Y, Li T, Li D, Zhang Y, Xiong W, Sun J, Tang Z, Chen G: **Using predicted shape string to enhance the accuracy of γ-turn prediction**. *Amino Acids* 2012, **42**:1749–55.

104. Sun J, Tang S, Xiong W, Cong P, Li T: **DSP: a protein shape string and its profile prediction server**. *Nucleic Acids Research* 2012, **40**:W298–W302.

105. Prati RC, Prati RC, Batista GEAPA, Monard MC: **Class Imbalances versus Class Overlapping: An Analysis of a Learning System Behavior**. In *MICAI 2004: Advances in Artificial Intelligence*. Springer Berlin Heidelberg; 2004:312–321.

106. Kubat M, Holte R, Matwin S: **Learning When Negative Examples Abound**. In *Machine Learning: ECML-97*. Springer Berlin Heidelberg; 1997:146–153.

107. Davis J, Goadrich M: **The relationship between Precision-Recall and ROC curves**. In *Proceedings of the 23rd international conference on Machine learning - ICML'06*. New York, New York, USA: ACM Press; 2006:233–240.

108. Hutchinson EG, Thornton JM: **PROMOTIF-a program to identify and analyze structural motifs in proteins**. *Protein Science* 1996, **5**:212–220.

109. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ: **Gapped BLAST and PSI-BLAST: a new generation of protein database search programs**. *Nucleic Acids Research* 1997, **25**:3389–402.

110. **PB-PENTAPEPT** [http://www.bo-protscience.fr/pentapept/?page_id=9].

111. Karatzoglou A, Wien TU, Smola A, Hornik K, Wien W: **kernlab – An S4 Package for Kernel Methods in R**. *Journal of Statistical Software* 2004, **11**:1 – 20.