Dissertation

A Study on the Protein Phosphorylation Site Prediction by a Set of New Features and Feature Selection with Grid Search

Graduate School of
Natural Science & Technology
Kanazawa University

Division of Electrical Engineering
and Computer Science

Student ID No.: 1424042020
Name: Lumbanraja Favorisen Rosyking
Chief Advisor: Professor Kenji Satou
Date of Submission: 30 June 2017

Abstract

Post-translational modification is one way of expanding genetic coding capacity to generate diversity in the corresponding proteomes. One of the most common post-translational modifications is phosphorylation. It is the process of adding a phosphate group to a target residue, which are Serine, Threonine, or Tyrosine.

Phosphorylation plays an important role in eukaryotic cell activities, such as cell cycle, signaling cell growth, and intracellular signal transduction. Research in the past has commonly conducted phosphorylation site identification using an experimental approach. One common experimental approach for identifying phosphorylation sites is by using mass spectrometry. By recording and measuring the mass of the ion sample, we can accurately identify phosphorylation sites. However, there are disadvantages in implementing mass spectrometry. (i) It requires an expensive machine. (ii) It also requires supporting tools and materials to conduct the experiment. (iii) Preparing the sample and analyzing it are both time consuming and labor intensive. (iv) Adequate skills are required to operate the machinery and analyze the results.

Another way to identify phosphorylation sites is the computational approach. A lot of researches implement this approach because of improvements in computer technology and machine learning. In general, there are two different methods of the computational approach. The first method is kinase-specific phosphorylation site prediction. It requires information about the protein kinase, which catalyzes the process, as well as information about phosphorylated protein sites. However, information about kinase proteins for phosphorylation is often not available publicly. The second method is the non-kinase-specific phosphorylation site prediction. This method only requires the information of the phosphorylated protein to conduct a prediction.

In this research, we conducted a non-kinase-specific phosphorylation site prediction by proposing new combinations of features. Feature selection was implemented to improve the classification result. There are two types of data sets we used to implement the method. The first data set is the P.ELM data set, which contains human and several animal phosphorylation sites. The second one is the PPA data set, which we used as an independent data set. This data set contains phosphorylation site information from plants. For each data set, we classified the phosphorylation in three different residues, Serine, Threonine, and Tyrosine. We implemented grid search to search the best number of features to achieve the highest classification performance.

Based on our experiment, creating new combinations of new features with features from previous research, and implementing feature selection can improve classification performance.

Comparing our results with the results of previous research, we can see an improvement of performance in phosphorylation site classification for Serine and Threonine residue.

Keyword: phosphorylation site, feature selection, grid search, classification

Acknowledgments

Studying and researching as a Ph.D student at Kanazawa University was an exciting and challenging experience for me. In these three years, many people were involved in helping and supporting my academic life. Without them, it would have been impossible for me to complete my Doctoral degree. Here is a small note of gratitude.

First of all, I would like to express my sincere gratitude to my professor, Professor Kenji Satou, for introducing me to the world of bioinformatics research. It was only because of his guidance, enthusiasm, and support that I could complete my research.

I am very thankful to all the committee members, Professor Kenji Satou, Professor Takeshi Fukuma, Associate Professor Yoichi Yamada, Associate Professor Hidetaka Nambo, and Associate Professor Makiko Kakikawa, for reading my thesis and giving me valuable remarks and suggestions.

I also would like to express my gratitude to the Indonesian Ministry of Research, Technology and Higher Education (RISTEKDIKTI) for sponsoring my scholarship. Because of this scholarship, I was able to achieve my dream of studying as a Ph.D. student.

I would like to thank Kanazawa University for providing me the opportunity to become a Ph.D. student here. My deepest gratitude to all the staff of Kanazawa University who have helped me in my academic life here in Kanazawa.

I would also like to thank my friends in the Bioinformatics Laboratory of Kanazawa University, for all the wonderful memories of working together and all the support. I would like to give my gratitude to my labmates, Vu Anh, Duc Luu, Ngoc Giang, Dau, Bahriddin, Reza, Bedy and Mera.

I am thankful for all my friends here in Kanazawa, especially colleagues from my university, Heri, Fitri, Trist, Ria, Bayu, Intan, and Rohaini. I also want to send my sincere gratitude to my brothers and sisters in Christ from Hope House, Shiro, Bodil, Gunn, Hannes, Harriet, and Kaoru.

Last but not least, I would like to thank my family, Mom, Dad, Mama Dominic, Lae Bapak Dominic, Mama Bonggas, Lae Bapak Bonggas, Romando, and the love of my life, Krystal. I am grateful for their love, prayers, understanding, and encouragement.

Thank you!

Contents

List of figures

List of tables

Chapter 1 Introduction

This chapter will explain several topics. First, it will introduce the background of this research, which includes protein translation and post-translational modification. It will also discuss about the process of protein phosphorylation in more detail. Then, it will explain the main objective and contribution of this research. Finally, it will explain how this thesis is organized.

1.1 Background

1.1.1 Protein translation.

Protein translation is the process by which a ribosome synthesizes a polypeptide string using the information from mRNA. Every three nucleotides (also known as a codon) in the mRNA is translated by tRNA into one amino acid. Figure 1.1 shows the process of protein synthesis in the cytoplasm cell. The ribosome attaches itself to the mRNA string and reads the nucleotide in the string. A tRNA containing three nucleotides (an anti-codon) that complement the codon of the mRNA will attach to the mRNA and then release the amino acid to the polypeptide string.

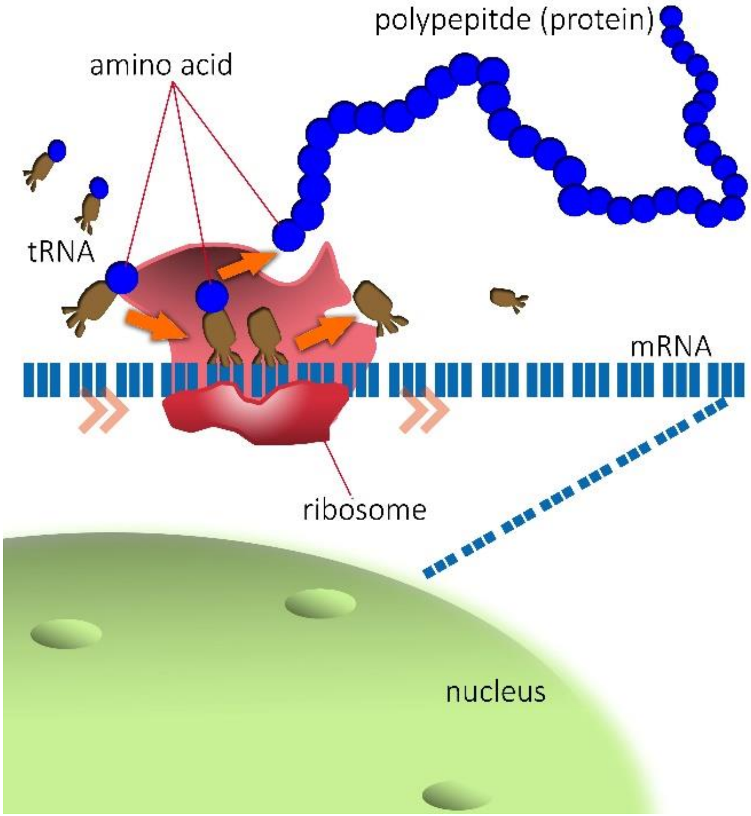


Figure 1.1 Diagram of protein translation from mRNA by ribosome

A polypeptide is a long string of 20 different types of amino acid attached together. This long string of amino acids is also known as the primary structure of the protein. As we can see in Figure 1.2, every amino acid consists of the same basic parts, which are an amino group, carboxyl, and a hydrogen atom. Only the side chain (R) is different in each amino acid.

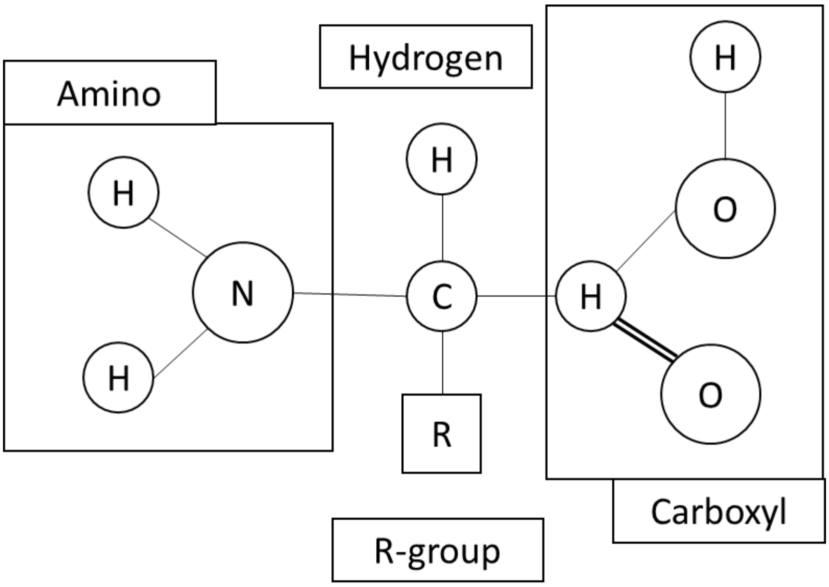


Figure 1.2 Structure of amino acid

The connection between one amino acid and another amino acid during the translation process is commonly known as a peptide bond. Figure 1.3 shows that the Amino acid (1) releases OH in the carboxyl part while attaching to the amino part of Amino acid (2), which releases a hydrogen atom. Since it creates a water molecule as a byproduct, this process is called the condensation process.

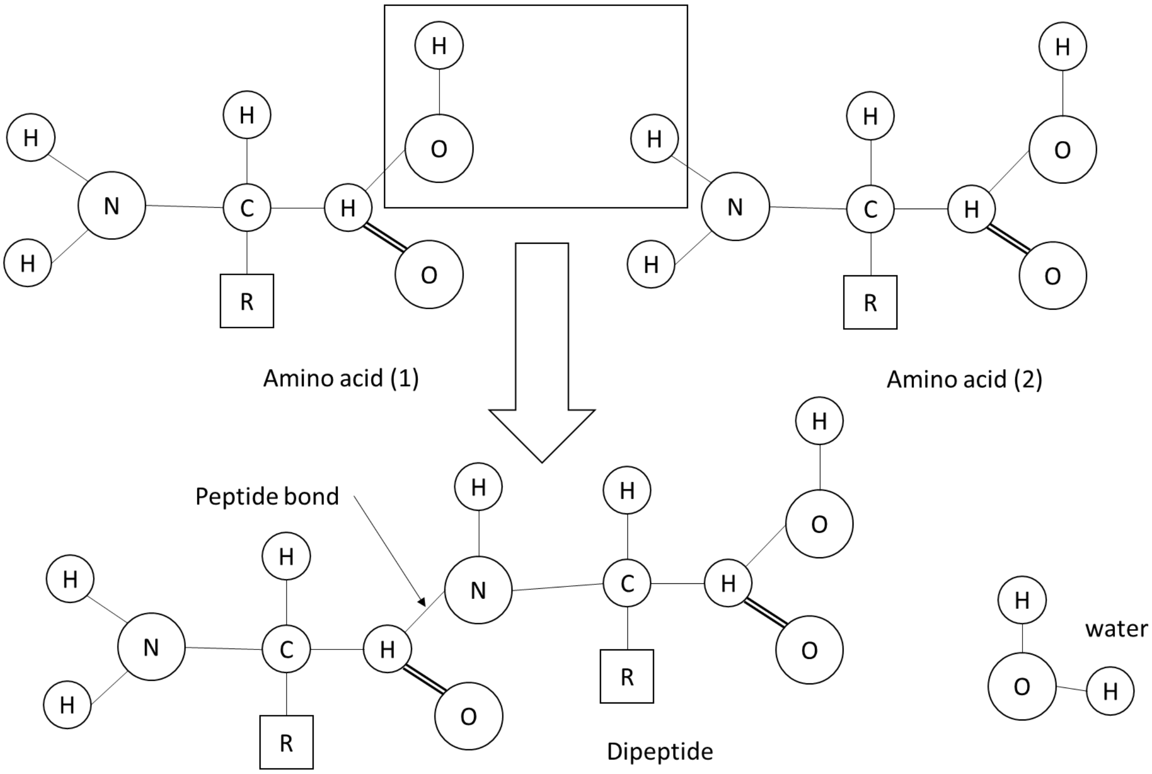


Figure 1.3 Diagram of peptide bond

Each amino acid in the polypeptide has different physicochemical properties based on the side chain; for example, Serine is hydrophilic and Valine is hydrophobic. The polypeptide string (also known as a backbone) will fold and twist, creating two common shapes, which are α-helix and β-sheet. These shapes are defined as a secondary structure. Figure 1.4 shows an output example of secondary structure prediction using Phyre2 [1]. It shows two forms, the amino acid string 'MIVRL' creates the β-sheet form, and the string 'GSKQAVDAAHKLM' creates the α-helix form.

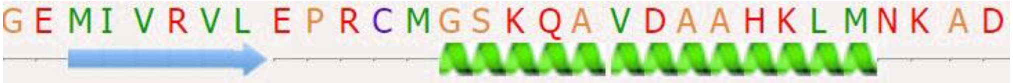


Figure 1.4 Secondary structure of protein prediction using Phyre2

Because of the physicochemical properties of the backbone, the protein will twist, bend, or fold, creating a more complex shape. This complex 3D structure is called the tertiary structure. Figure 1.5 shows an example of a protein's tertiary structure (source: http://brussels-scientific.com/wp-content/uploads/2016/08/RNase_A.png).

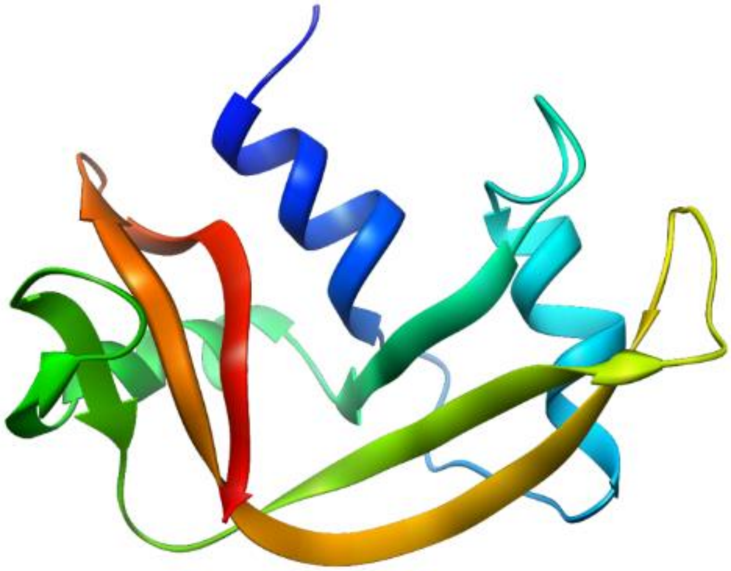


Figure 1.5 Example of protein tertiary structure (source: Gerard T., 2106)

1.1.2 Post-translational modification

Many proteins are modified after protein translation completed, which is known as Post-translational modification (PTM). PTM occurs when the protein interacts with a specific enzyme-catalyzed modification on the backbone or side chain. Commonly, this process happens in several places in the cell, for example in the cytosol, endoplasmic reticulum, or Golgi apparatus.

PTM is one way of expanding the genetic coding capacity to generate diversity in the corresponding proteomes, as is shown in Figure 1.6 [2]. PTM cellular regulation is complex and plays a very important role in biological regulation. It helps the cell regulate localization, cellular activities, and interaction with other cellular molecules.

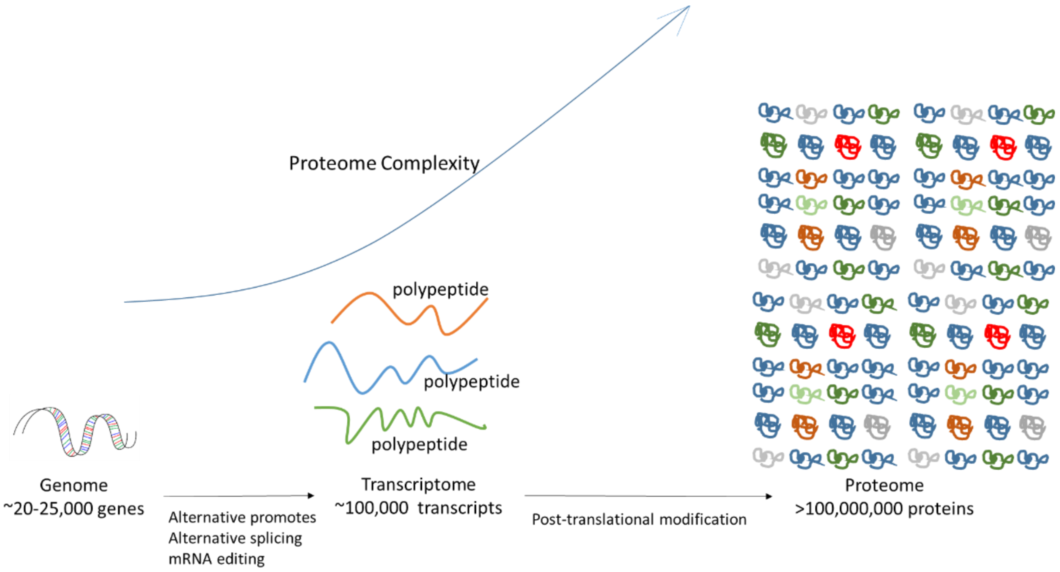


Figure 1.6 Comparison of complexity from genome to proteome (source: Thermofisher Scientific)

There are different types of PTM. These are the common ones:

- **Methylation**. Methylation is the process of transferring one carbon methyl group to amino acid side chains by methyltransferases using S-adenosylmethionine (SAM) as the primary methyl donor. This can neutralize a negative amino acid charge when bound to carboxylic acids, and leads to an increased hydrophobicity in the protein. A well-known purpose of methylation is epigenetic regulation of transcription.
- **Acetylation.** Acetylation, specifically to nitrogen atoms on a protein (N-acylation). This occurs as the nascent protein is being translated. The N-terminal methionine on the growing polypeptide chain is cleaved by the methionine amino acid peptidase and then released by an acetyl group donated by acetyl CoA via enzyme N-acetyltransferase. Around 90% of eukaryotic cells are acetylated using this process.
- **Glycosylation**. Glycosylation involves the addition of various types of sugar moieties. It ranges from a simple monosaccharide modification of transcription factors, to highly complex branched polysaccharide modification of cell surface receptors. These carbohydrates can be added to the nitrogen atom in the side chain of asparagine residues, which are called N-linked. Another type of Glycosylation is the addition of oxygen atoms in the side chain in Serine or Threonine residues, which are called O-linked. These types of glycosylation changes create the structural component of cell surface and secreted proteins.

- **Lipidation**. Lipidation is a PTM which often occurs in particular membrane-bound organelles, such as the endoplasmic reticulum, Golgi apparatus, or mitochondria. It is also used to target proteins to endosomes, lysosomes, and the plasma membrane. There are two types PTK lipidation, GPI anchors, and S-palmitoylation. C-terminal glycosylphosphatidylinositol (GPI anchor) helps to tether proteins bound to the plasma membrane of the cell surface. These hydrophobic moieties are prepared in the endoplasmic reticulum, where they are added to nascent proteins and used to localize cell surface proteins to cholesterol, or sphingolipid-rich areas in the plasma membrane. S-palmitoylation involves the addition of 16 carbon long paimitoyl groups to dilate side chains of cysteine residues. This modification adds a long hydrophobic chain that can be used in a similar manner as a GPI anchor. It helps to anchor proteins in the hydrophobic cell membrane.
- **Ubiquitination**. Ubiquitination is a PTM used to target proteins for degradation. Ubiquitin is a polypeptide consisting of 76 amino acids, which attaches to lysine residues of target proteins via the C-terminal glycine of ubiquitin. Polyubiquinated proteins are recognized by the 26th proteasome, which is an enzyme that catalyzes the degradation of the protein and the recycling of the ubiquitin.
- **Proteolysis**. Proteolysis is a PTM which uses proteases to remove amino acids from the amino end of the protein, or to cut the peptide chain in the middle. One example of proteolysis is the peptide hormone insulin, which is cut twice after disulphide bonds are formed. Furthermore, a pro-peptide is removed from the middle of the chain. The resulting protein consists of two polypeptide chains connected by disulphide bonds. Proteases also plays a role in cell signaling, antigen processing, and adaptosis.

Among all the PTMs that occur in eukaryotic cell, one of the most common is phosphorylation.

1.1.3 Phosphorylation

Protein phosphorylation is a reversible modification of adding a phosphate group to certain residues, which are Serine, Threonine or Tyrosine [3]. It is used to regulate proteins in various cellular processes, including signal transduction pathways, the cell cycle, and apoptosis. Protein kinases are the enzymes that help facilitate the phosphate group transfer and phosphorylases help to remove them.

As shown in Figure 1.7, this process includes the transfer of a phosphate group from Adenosine Triphosphate (ATP) to the target residue (Serine, Threonine, or Tyrosine), thereby creating Adenosine Diphosphate (ADP) as the byproduct. This PTM event normally occurs in the cytosol or the cell nucleus. The kinase protein helps the phosphorylation process, which has an important role in regulating cellular activities, such as metabolism, proliferation, differentiation and apoptosis. Most families of the kinase enzymes have the same homologous catalytic domains and the mechanism of substrate recognition may be similar despite the wide scope of variation in sequence.

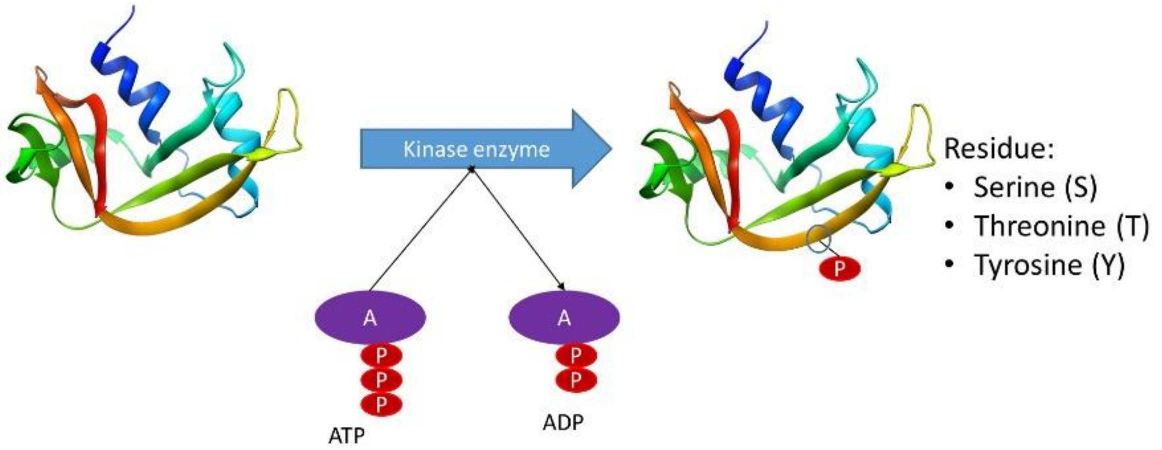


Figure 1.7 Process of protein phosphorylation

1.2 Objective

Protein phosphorylation has an important role in eukaryotic cell activities, which include the life cycle of the cell, signaling for cell growth, and intracellular signal transduction. This is a big reason why a lot of researches are conducted to analyze and predict phosphorylation sites. The main objective of this research is to find new combinations of features and selecting important features to improve the performance of phosphorylation site classification using the computational approach.

1.3 Contribution

Protein phosphorylation is one of the most common types of post-translational modification, and it is important for the cell. Studies related to phosphorylation site prediction using different methods have been explored intensively by researchers.

This research may contribute in the following matters:

Purposing new features for phosphorylation site prediction. We propose several new features, which have not been used to conduct classification previously. We generated these features with several state-of-the-art protein analysis tools.

Finding combinations of new features with features that have been used for the classification method. We combine new features with features that have been used from previous methods to achieve a better classification performance.

Implementing feature selection to improve classification performance. In previous research, feature selection was implemented. However, it decreased the performance of classification. In this research, we conducted feature selection using the combination of new features and features from previous research to improve the classification result.

1.4 Thesis organization

This thesis is divided into five chapters.

Chapter 1 introduces the background of this research topic and the reasons of conducting the research.

Chapter 2 explains the most recent literatures of protein phosphorylation site prediction. They include different approaches for prediction methods. Several feature selection and classification methods will also be listed and explained. Finally, in this chapter we will also explain about cross validation.

Chapter 3 introduces the data sets which were used for classification. This includes information about the data sets and how we prepared the data sets to conduct classification. In this chapter, we also explain about the novel features, feature selection, and classification methods. Finally, we explain about evaluation metrics and grid search, which we used to search and evaluate the best classification performance.

Chapter 4 shows and explains the result of our experiments in detail. This includes feature selection and classification results. Comparison of results with previous research related to this topic is explained in this chapter.

Chapter 5 summarizes the thesis by stating a conclusion of achievements. Suggestions for the future work are discussed in this chapter.

Chapter 2 Literature review

This chapter will explain and discuss about several approaches of phosphorylation site prediction, which include the experimental approach and computational approach. Two related research methods, which are PhosphoSVM and RF-Phos will also be discussed. We will also explain about feature selection, classification, and cross validation.

2.1 Phosphorylation site identification

There are two common approaches in identifying protein phosphorylation sites. They are the experimental approach and the computational approach.

2.1.1 Experimental approach: Mass spectrometry

In the past, researchers relied on the experimental approach to analyze protein and identify its phosphorylation sites. One common method has been to use a machine called (as shown in Figure 2.1) a mass spectrometry (MS) machine.

Figure 2.1 Mass spectrometry machine (Source: Business Wire, 2014)

Mass spectrometry is a method of splitting an atom, isotope, or even fragmented molecules based on their respective masses. Generally, a MS machine consists of three parts, which are an ionizer, mass analyzer, and a detector, as shown in Figure 2.2. The Ionizer is a vacuum where the sample is input. The sample is hit by electrons and several positive ions are created. The mass analyzer consists of two components, which are an electric field and a magnet. The electric fields consist of negative ions that will pull the positive ion to the mass analyzer. In addition, the magnets will bend the path of ions. Finally, the detector consists of an electro multiplier and amplifier.

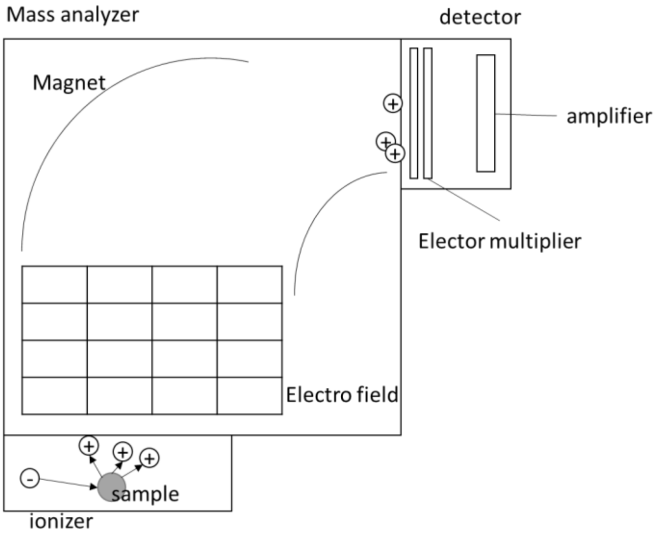
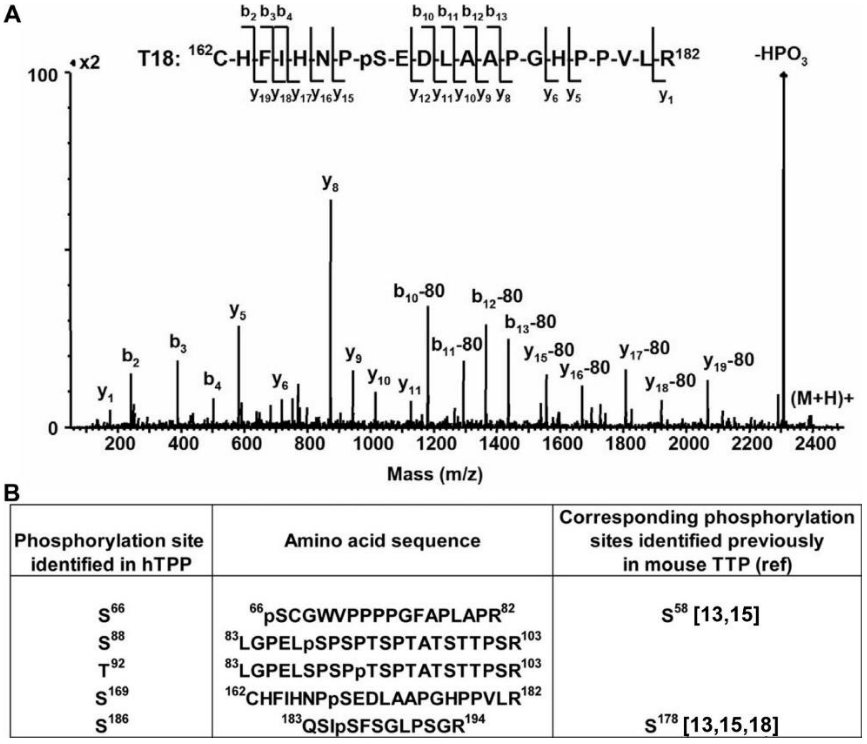


Figure 2.2 Diagram of a mass spectrometry machine

Calibration is required to be conducted before using a MS machine. The strength of the magnets must be calibrated in order for the positive ions from the sample to be received by the electron multiplier in the detector. Once calibrated, a sample is hit by a negative electron, this releases the positive ions from the sample. Using the negative charge in the electric field, the positive ion moves to the mass detector. The magnet is then used to bend the path of the positive ions. Heavier ions are harder to move than lighter ones. The electron multiplier then catches the positive ions and the result is amplified using the amplifier. We can identify the ions based on where the ion is located; the ions with heavier masses will be located higher than the ions with lighter masses.

Based on that concept, we can also analyze complex molecules such as protein sequences. Amino acids can be identified by their masses. MS-based proteomics is commonly known as an indispensable technology for interpreting information encoded in genomes. Currently, protein analyses, especially PTM by MS, has been most successful when conducted on data sets that consist of small protein sequences isolated in a specific context [4].

Cao conducted research to identify phosphorylation sites using MS [5]. Figure 2.3 shows an output from MS to identify phosphorylation sites at Serine^{66}, Serine^{88}, Threonine^{92}, Serine^{169}, and Serine^{189}.



Phosphorylation site identified in hTPP	Amino acid sequence	Corresponding phosphorylation sites identified previously in mouse TTP (ref)
S^{66}	^{66}pSCGWVPPPPGFAPLAPR^{82}	S^{58} [13,15]
S^{88}	^{83}LGPELpSPSPTSPTATSTTPSR^{103}	
T^{92}	^{83}LGPELSPSPpTSPTATSTTPSR^{103}	
S^{169}	^{162}CHFIHNPpSEDLAAPGHPPVLR^{182}	
S^{186}	^{183}QSIpSFSGLPSGR^{194}	S^{178} [13,15,18]

Figure 2.3 Example of mass spectrometry to identify phosphorylation at Serine (Source: Cao et al., 2006)

The main advantage of using MS is that it can produce a high accuracy in phosphorylated protein site identifications on a mass spectrometer reading. However, it also has disadvantages. It requires expensive equipment. According to LabX website, a compact size MS type Shimadzu GZ/MS system (including computer and software) costs around $30,000 US dollars. Secondly, to conduct this experiment, it requires other supporting equipment and materials, such as a centrifugal machine. It also requires intensive labor and preparation time. Extracting protein sequences from samples requires pre-processing the sample which is also time consuming. Finally, not everyone can conduct this research to identity phosphorylation sites. It requires adequate skills and knowledge to prepare the sample and operate the machine and software.

2.1.2 Computational approach

Currently, because of the advancement of computer and information technology, researchers more commonly use computer technology to identify phosphorylation sites. There are four basic reasons why the computational approach is becoming more popular. First, a new generation of high-speed computer processors with multi-core and multi-thread technology have been released. Second, large data storage is becoming more affordable. Third, there are new computer networking technologies that make data transfer faster and more reliable. Forth, new machine learning algorithms that make computers able to solve complex problems are being developed.

In general, phosphorylation site prediction using the computational approach can be divided into two methods, which are the kinase-specific approach and the non-kinase-specific approach.

i. Kinase-specific approach

To conduct phosphorylation site prediction using this approach, two areas of information are required. First is the information about the kinase protein, which catalyzes the phosphorylation. For example, the kinase family in Homo sapiens are AGC kinases, CaM kinases, CK1, CMGC, STE, TK, TKL. Second is information about the protein target of phosphorylation, including the information of residue that has been phosphorylated.

There have been several research works conducted using this approach. Xue et al, proposed a method called GPS 2.1 using JAVA 1.5 [6]. Motif length selection (MLS) was implemented to improve the prediction of the previous method (GPS 2.0). In this work, they use phosphorylation site data from humans and several species of animal. Information about human protein kinase is also collected and classified into four groups.

Bloom introduced NetphosK [3]. This method implemented Neural Network for the classifier. In this research, they selected different types of protein kinase, which are PKA, PKC, PKG, cdc2,CL-2, and CaM-II. They also collected information about phosphorylated Serine and Threonine, which are catalyzed by protein kinase.

The main problem of implementing this approach is that kinases protein information is typically not publicly available.

ii. Non-kinase-specific approach

This approach only requires information about the protein targets of phosphorylation, including phosphorylated residue. Many computational techniques using this approach have been implemented for phosphorylation site prediction. In this thesis, two related works using this approach will be explained.

2.2 Related works

2.2.1 PhosphoSVM

PhosphoSVM was introduced by Dou in 2014 [7]. This method implemented eight different feature groups to classify phosphorylation sites. It implemented Support Vector Machine (SVM) for the classifier. The feature groups are named Shannon Entropy, Relative Entropy, Secondary Structure, Protein Disorder, Accessible Surface Area, Overlapping Properties, Average Cumulative Hydrophobicity, and K-Nearest Neighbor Profile.

In the paper, classification was conducted with two different data sets: the P.ELM (human and animal) data set, and the PPA (plant) data set as the small independent data set. The AUC values for the P.ELM data set are: 0.84, 0.82, and 0.74 for Serine, Threonine, and Tyrosine, respectively. In addition, for the PPA data set, the AUC values for Serine, Threonine, and Tyrosine are: 0.74, 0.67, and 0.60, respectively. Feature selection was not implemented in this method.

2.2.2 RF-Phos

Ismail proposed his method: RF-Phos in 2016 [8]. Ten different feature groups were used to conduct classification. Features that RF-Phos used from PhosphoSVM are: Shannon Entropy, Relative Entropy, Accessible Surface Area, Overlapping Properties, and Average Cumulative Hydrophobicity. This method also introduced new features, which are Information Gain, Sequence Feature, Composition-Transition-Distribution, Sequence Order Coupling Numbers, and Quasi Sequence Order.

Using those features, Random Forest was used to classify the phosphorylation sites. The primary data set was P.ELM, and the independent data set was PPA. This method achieved a better performance when compared to PhosphoSVM. The accuracy for Serine, Threonine, and Tyrosine using the P.ELM data set were 0.83, 0.87, and 0.86, respectively. Also Random Forest was used to conduct feature selection. Gini Impurity Index (GII) was proposed to measure the important features. In the research, the results of classification using only the top 100 important features was compared with the results of classification using all the features. In general, it was found that feature selection using only the top 100 important features decreased the classification performance.

2.3 Feature selection

In real-world situations, our data contains relevant and irrelevant information. However, relevant and irrelevant features for many real-world learning problems are often unidentified. The problem with data sets containing irrelevant information is that it could degrade the performance of classification, both in computational time (because of high dimensional data) and in accuracy of prediction (because of irrelevant information). Therefore, it is important to identify and select relevant features. Feature selection is a process of selecting relevant feature subsets. There are several important reasons for implementing feature selection, to help visualize and understand the data, reduce data storage, reduce computation time, and break the curse of dimensionality in order to improve classification performance [9].

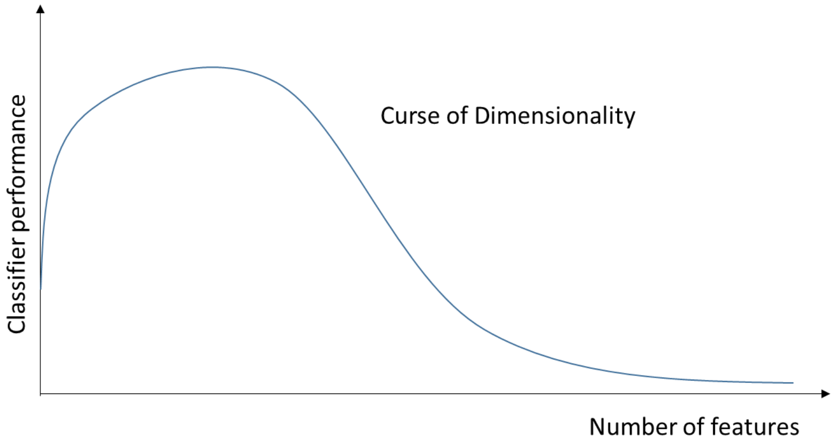


Figure 2.4 Graphic illustrating the curse of dimensionality

Figure 2.4 illustrates the curse of dimensionality. This occurs in a classification or prediction method that uses data containing a very large number of features. The performance of classification reduces as the number of features used increases.

This method is used to select a sub group of features in order to improve the performance of the classifier. In other words, given a feature set $F = \{x_1 x_2, \dots, x_n\}$, the goal of this method is to find a subset F' that maximizes the learning ability classifier. However, it would not be practical to implement a brute force approach to search each possibility from the large number of features. If our data contains *n* features then there will be a 2^n possibility of finding feature subsets.

2.3.1 Wrapper method

This feature selection method was introduced by Kohavi, 1997 [10]. The goal of this method is to evaluate how useful a feature set is by using a learning algorithm. For this method to be able to select a subset of features, a learning model is trained for each different feature subset. The selected subset is the one that has the best learning performance.

There are several requirements for implementing this feature selection method. First, it requires the measurement method to evaluate the selection performance. Performance measurement is implemented to generate the criteria of feature selection and to create the resampling strategy. Second, it requires a learning method. Finally, a method that is able to search all possible feature subsets is necessary.

There are two terminologies used in the search method. They are forward selection and backward elimination. Forward selection can be define as the process of searching from the empty feature set. Backward elimination is the process of deleting from a full feature set. In most experiments, the initial state is set to be empty, therefore forward selection is most commonly implemented. The main reason is because of computational time. It requires less time to generate classifiers for a small number of features. However, theoretically, by using a backward elimination we can search all features easily.

2.3.2 Filter method

Filter method conducts feature selection by using an attribute evaluator and an algorithm ranking system to rank all the features in a data set. This generates a list of features and their given ranks, in association with attribute evaluation. By omitting one feature at a time from the list provided by the algorithm ranking system, we can evaluate the performance of the features with a classification algorithm.

A disadvantage of this method is that the value from the algorithm ranking system may be different from the value given by the classification algorithm. This may cause the model to be overfit.

2.4 Classification

Classification is a process using collected data to assign discrete labels. The goal is to predict the class of new observations. Classification tries to generate a classifier than can produce an output from arbitrary input. Classifiers can then label and assign an unseen example into a specific class. There are two main characteristics of classification problems. First, the output of classification is qualitative. Second, the classes to which a new observation can belong are known beforehand.

In general, there are two classification problems. First, the binary-class classification has only two class labels. Second, the multi-class classification has more than two class labels.

The possible applications of classification methods are very broad. For example, after a set of clinical examinations that verify the vital signals of a disease, we can predict whether a new patient with an unseen set of vital signals suffers that disease and needs further treatment. Another example is classifying a set of animal images into their species label.

2.4.1 Decision tree

This is a simple method for solving classification problems. The objective of this method is to generate a binary tree, which minimizes the error in each leaf. The main advantage of a decision tree is that it is easy to read and understand, as illustrated in Figure 2.5. In this data set, there are two class labels, which are *A* and *B*. This data set consists of two data variables x_{i1} and x_{i2}. The leaves in the tree represent class labels and the nodes represent the conditions that lead to the class labels.

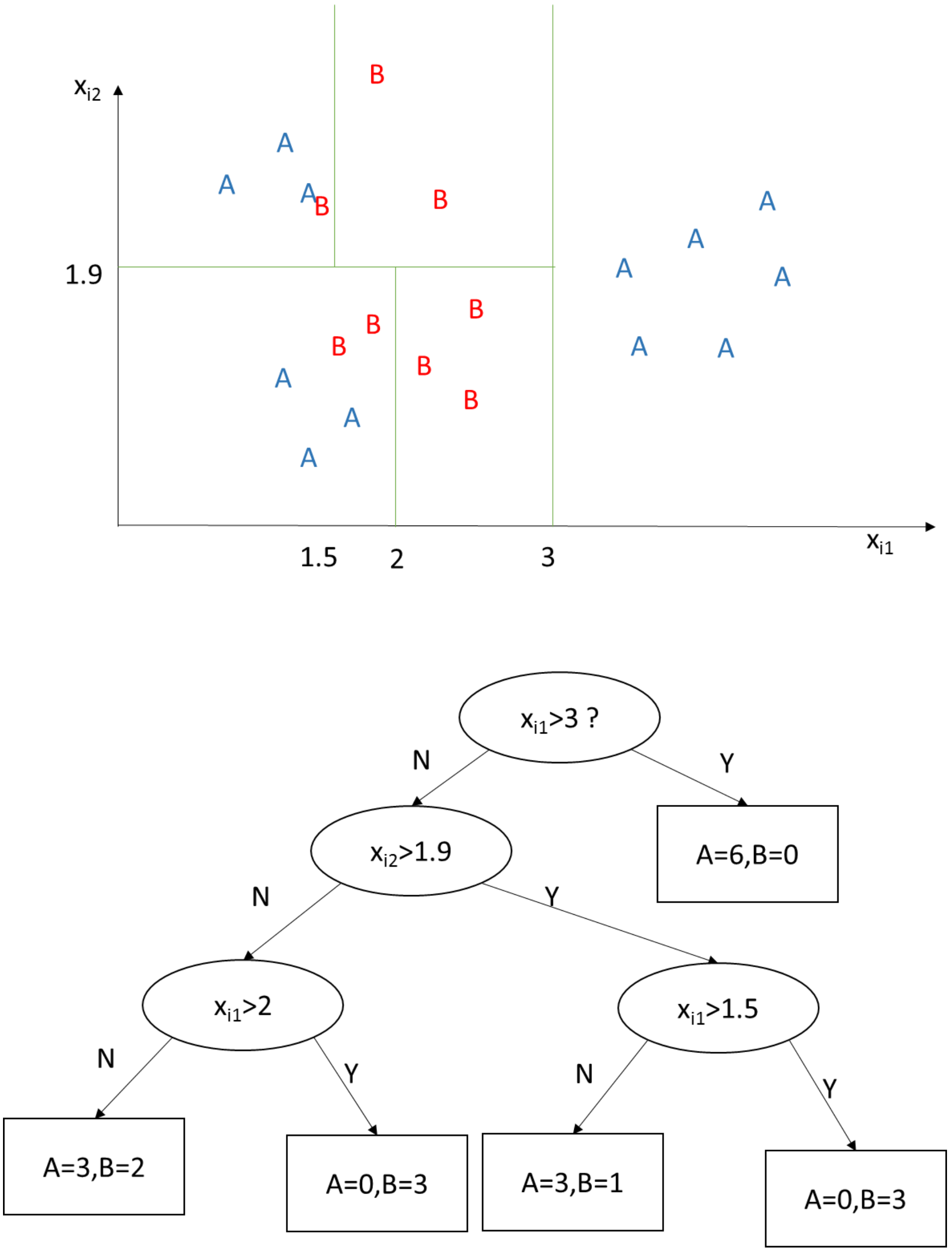


Figure 2.5 Illustration of decision tree using two variables.

To build a decision tree we use data to determine several points. First, we have to decide which variable is used to split at a node and what will be the value of the split. The basic idea is to find a condition that will split class labels in a way that creates groupings consisting of the maximum possible number of identical class labels. To measure the split performance, we use a method called entropy. Second, we need to determine when to stop (create a leaf) or split again. Finally, we have to assign a leaf to the class labels

2.4.2 Random Forest

Introduced by Breiman, this is one of the more popular classification methods [11]. This method generates many decision trees based on random selection of data and random selection of features. It provides classes of dependent features on the various trees. Figure 2.6 illustrates classification using Random Forest. The subsets are selected randomly so that they consist of different numbers of data and features.

From the randomly selected subset of data, we create different decision trees. There are two reasons why we have to generate features randomly. First, most of the tree can generate a correct classification of class for most of the data set. Second, error generated in each tree occurs in different places. By conducting voting for each observation and deciding about the predicted class based on the voting result, this method is expected to have a better classification result.

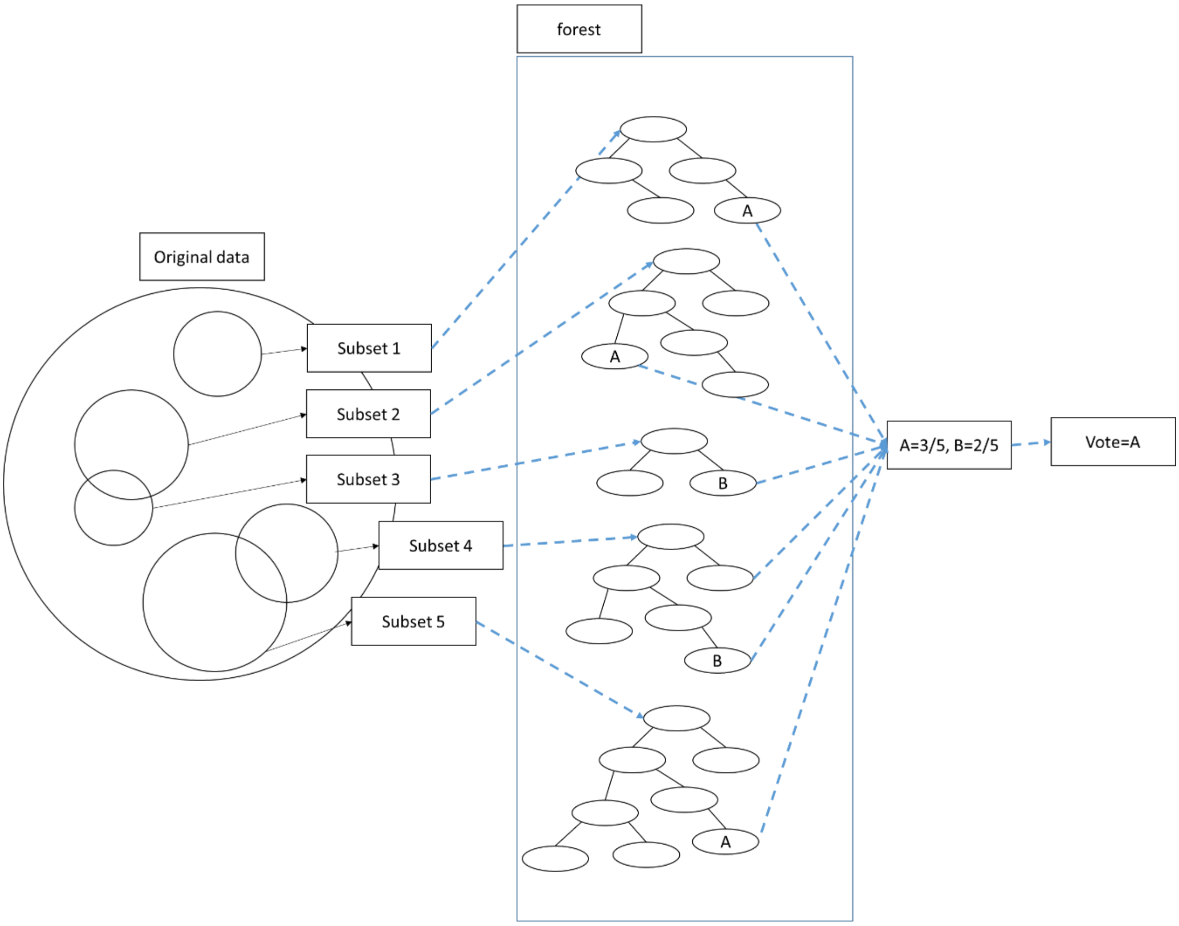


Figure 2.6 Classification process using Random Forest

2.4.3 Support Vector Machine

Another one of the more popular classification methods is the Support Vector Machine (SVM). It is proposed by Vapnik [12]. SVM models the classification problem by creating a feature space, which is a finite-dimension vector space, each dimension of which represents a feature of a particular object. In other words, SVM constructs linear separating hyperplanes in high-dimensional vector space. Data points are defined as $(\vec{x}, y)$ tuples, $\vec{x} = (x_1, x_2, \dots, x_p)$ where x_j are the feature values and y is the class label. Optimal classification occurs when the hyperplane separates with maximum distances to the nearest training data set point, as illustrated in Figure 2.7. In this example, two classes are separated using a linear hyperplane.

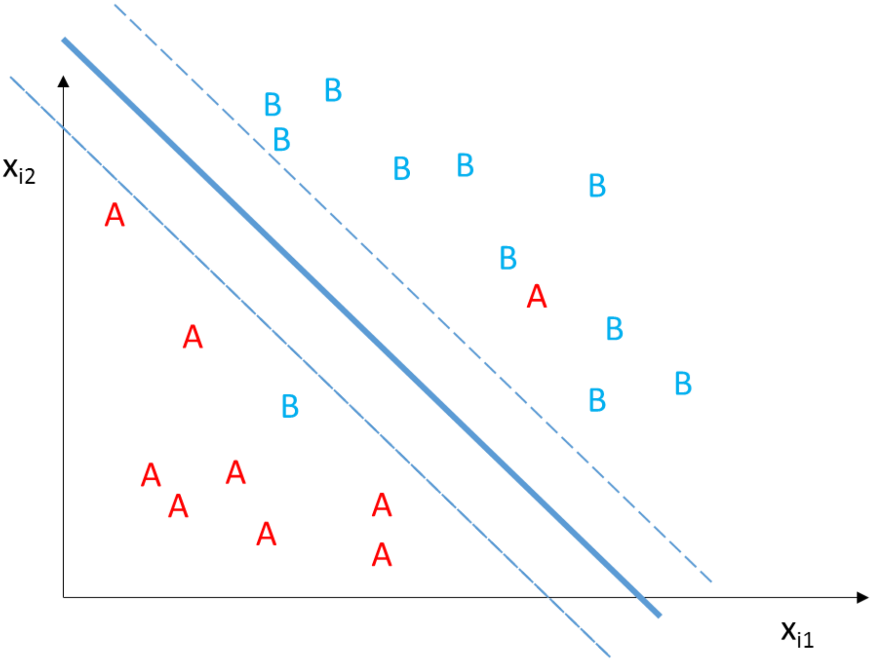


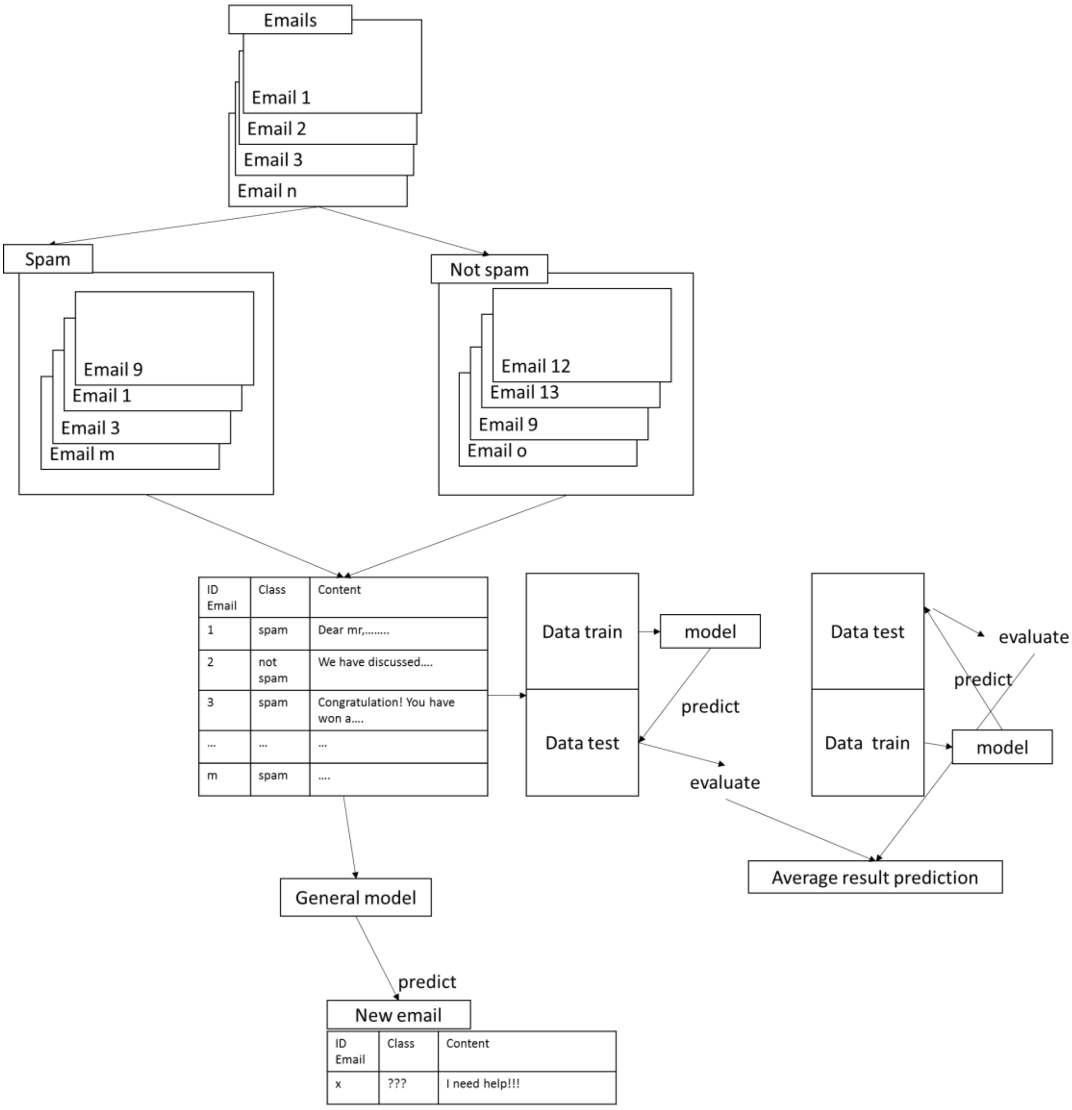
Figure 2.7 Linear hyperplane classifying two classes

SVM has many advantages, mostly because of its computational efficiency on large data sets. The first advantage of SVM is that it is an efficient classifier in high-dimensional spaces. This is particularly applicable to text or DNA/protein sequence classification problems where the dimension of the data set can be extremely large. Secondly, it is memory efficient. Since only a subset of the training data set is used in the actual process of assigning new members to a class, only this subset needs to be stored in the memory when making classification decisions. Thirdly, it is versatile. Separation of classes is often non-linear. The ability to implement different kernels allows flexibility for decision boundaries, leading to a better performance.

2.5 Cross validation

Cross validation is a method used to evaluate prediction performance from a certain model. The main concept of this method is to split the data set into training data and testing data. This is done to avoid overfitting the result and create a generalizable prediction model. The model is created by using the training data, and the test data is used for evaluating the performance of prediction. In addition, we hope that the model is generalizable enough to predict class labels from data that the model has not seen before.

For example, we can use this method to create a system that can detect a spam email, as it is illustrated in Figure 2.8. First we collect the data of all the emails and we set a label called ‘spam’ or ‘not spam’ for each email. Then we split the data into training and testing data. We create the classification model using the training data and evaluate it with the test data. The result of prediction is then compared with actual class label. We then change the role of each subset of data. The test data from the previous step becomes the new training data, and vice versa with the training data from the previous step. We calculate the accuracy of the prediction using the model generated from training data. We then average the accuracy of both testing processes.



ID Email	Class	Content
1	spam	Dear mr,........
2	not spam	We have discussed....
3	spam	Congratulation! You have won a....
...	...	...
m	spam	....

ID Email	Class	Content
x	???	I need help!!!

Figure 2.8 Example of email spam prediction. Cross validation is used to test the model.

2.5.1 *k*-fold cross validation

One common implementation of k-fold is where k=10. Figure 2.9 describes the implementation of 10-fold cross validation. First, the data set is divided into ten groups. Ten iterations of cross validation are conducted for all groups, where 90% of the data is used to create the model to test 10% of the data. Then the average result of all iterations is used to measure the performance of the classification using the data set.

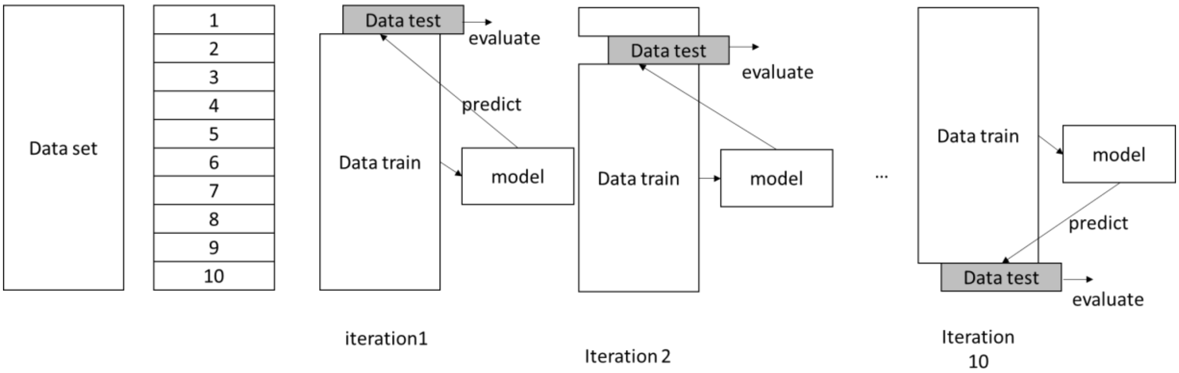


Figure 2.9 Procedure of 10-fold cross validation

2.5.2 Leave-One-Out cross validation

An extreme example of k-fold cross validation is Leave-One-Out cross validation. In this setting, we take one sample and leave it out and we generate the model based on the rest of the data set. After that, we use the model to evaluate the test data. We repeat this process for each data in the data set, as it is illustrated in Figure 2.10. In this example, the data set consists of only 4 objects of observation. We split the data set into 4 folds, taking one out for the test data. Using the 3 data sets left, we create a model for predicting the test data. This method is commonly used when the data set is not large, especially in the biomedical field where there are only a very small number of samples available for the data set.

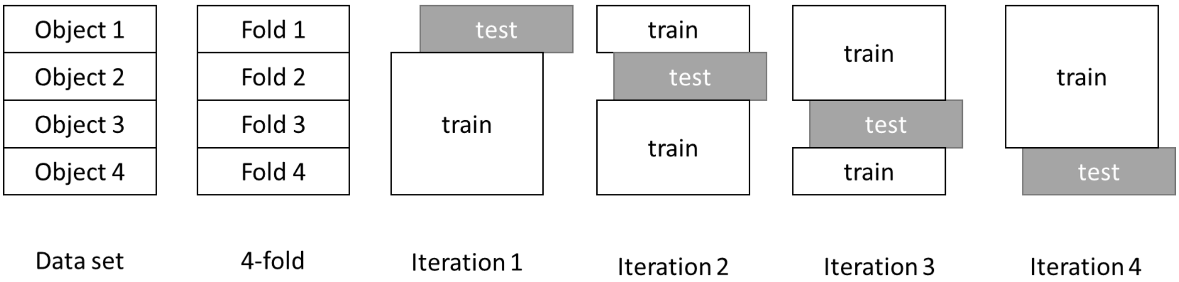


Figure 2.10 Illustration of LOOCV. In this example, there are 4 objects in the data set.

Chapter 3 Data and method

This chapter will explain the two data sets, which are used in this research. The flowchart of the research will also be explained. Each process in the method will be explained in detail. The method of finding feature sets which achieve the best classification performance will also be explained.

3.1 Data

In this research, we use two different data sets, which are P.ELM and PPA.

3.1.1 P.ELM data set

P.ELM is a database containing phosphorylation sites in the eukaryotic cell which have been experimentally verified [13]. The database consists of 42,574 phosphorylation sites, which are: 31,754, 7,449, and 3,370 for Serine, Threonine, and Tyrosine, respectively. Most of the information is from: *Homo sapiens* (62%), *Mus musculus* (16%), *Dorsophila melanongstar* (13%), *Caenorhabditis elegans* (7%). This data set was collected by Dou and redundant sequences with 30% similarity were removed, as shown in Table 3.1. The data was made available for download from the web site of PhosphoSVM [7].

Table 3.1 P.ELM data set of phosphorylation sites for Serine, Threonine, and Tyrosine residue

Residue	Number of Sequences	Number of Sites
Serine	6,635	20,964
Threonine	3,227	5,685
Tyrosine	1,392	2,163

We then created protein sequences that have fixed-lengths. The window size for these sequences is 9, with the phosphorylatable residue (Serine, Threonine, or Tyrosine) located at the center. A sequence was defined as 'positive' when the center of that sequence is a known phosphorylated residue; otherwise, it is defined as a 'negative' sequence. We removed redundant sequences for both positive and negative sequences by using skipredundant [14]. The parameters we implemented using skipredundat are as follows: the acceptable percentage of similarity was set to 0-20%, the value for gap opening penalty to 10, and gap extension penalty to 0.5. Table 3.2 lists the number of positive and negative sequences before and after removing redundant sequences for each residue. The number of negative sequences after redundancy removal are: 4,771, 3,343, and 898 for Serine, Threonine, and Tyrosine, respectively. We then selected negative sequences randomly for each residue based on the negative sequences from Ismail's work.

Table 3.2 Number of sequences before and after removing redundant sequences for window size-9

Residue	Positive		Negative
	Before	After	
Serine	20,557	1,554	1,543
Threonine	5,596	707	453
Tyrosine	1,392	267	226

3.1.2 PPA data set

The second data we used was PPA, as a small independent data set. PPA is a database containing phosphorylation sites from *Arabidopsis thaliana* [15]. We created protein sequences for this data set using the same window size and method as P.ELM. After removal of redundant sequences, we selected positive and negative sequences randomly also based on Ismail’s work. We can see in Table 3.3 the number of positive and negative phosphorylation sites for each residue with window size 9. We set the number of positive and negative sequences as equal in order to make the data set well balanced.

Table 3.3 PPA data set, the independent data set

Residue	Number of positive/negative sequences after redundancy removal	Number of positive/negative sequences after selection
Serine	484/1830	307/307
Threonine	132/1227	68/68
Tyrosine	187/640	51/51

3.2. Method

3.2.1 Flowchart of research method

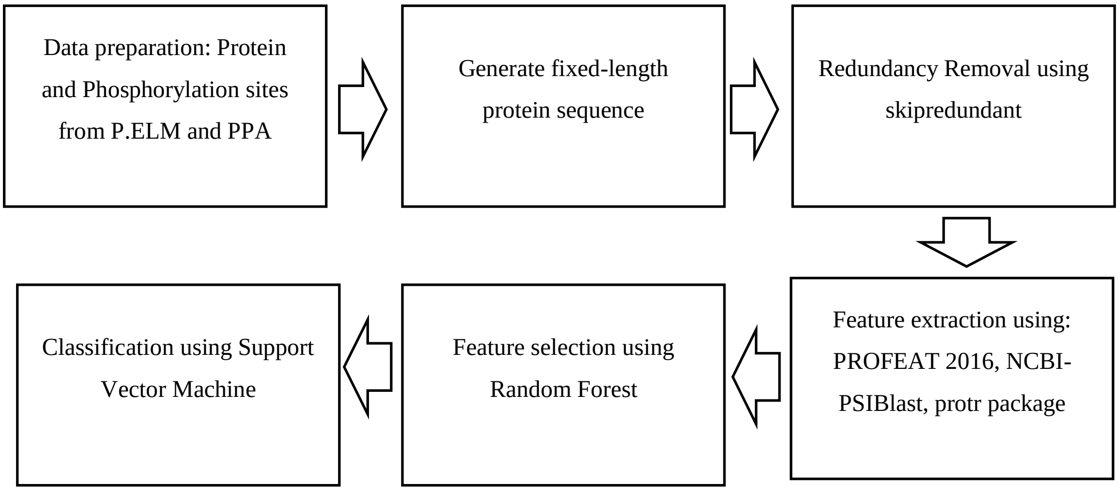

Data preparation: Protein and Phosphorylation sites from P.ELM and PPA
Generate fixed-length protein sequence
Redundancy Removal using skipredundant
Feature extraction using: PROFEAT 2016, NCBI-PSIBlast, protr package
Feature selection using Random Forest
Classification using Support Vector Machine

Figure 3.1 Flowchart of the research method

We conducted six processes in our research, as shown in Figure 3.1. First, we collected the data of proteins related to phosphorylation and the position of the phosphorylated residues from the P.ELM and PPA data sets. Then we generated fixed length sequences. To reduce the computational time and create a non-redundant data set, we removed similar protein sequences using skipredundant. Then we generated features from the protein sequence using PROFEAT 2016, NCBI-PSIBlast, and protr package. We then conducted feature selection using Random Forest. Finally, we classified phosphorylation sites for each residue. We found the best feature selection by implementing grid search. In this research, we compared our results of classification after feature selection with the results from other works related to phosphorylation site prediction.

3.2.2 Feature extraction

Feature extraction generates a series of features by analyzing the original data. Using a fixed-length protein sequence, we implemented feature extraction to generate information as numerical vectors. The features that we used in this research were extracted using three tools: PROFEAT 2016, NCBI-Psiblast, and protr package.

PROFEAT (2016) is a web server that provides tools to extract features related to proteins from a list of protein sequences [16]. This web server is used to analyze and predict structural, functional, expression, and interaction information of proteins (polypeptides). We used it to generate the following features: Amino Acid Composition (AAC), Dipeptide Composition (DPC), Normalized Moreau-Broto Autocorrelation Descriptor (NMB), Moran Autocorrelation Descriptor (MORAN), Geary Autocorrelation Descriptor (GEARY), Composition, Transition, Distribution Descriptor (CTD), Amphiphilic Pseudo-Amino Acid Composition (APAAC), and Total Amino Acid Properties (AAC).

Position-Specific Iterative (PSI)-BLAST is a search method based on a protein sequences profile that creates alignments generated by running BLASTp (protein) program [17].

protr is an R package that provides tools to generate various numerical information from a protein (polypeptide) sequence [18]. This package generates eight different feature descriptor groups. From these eight groups, generally around 22,700 descriptor values are implemented. This package also allow the user to select amino acid properties from AAIndex database, and other properties that the user can define to generate customized descriptors. protr is used to produce the following features: BLOSUM and PAM Matrices for the 20 Amino Acids, Amino Acid Properties Based Scales Descriptor (Protein Fingerprint), Scales-based Descriptor derived by Principal Components Analysis, Scales-based Descriptor derived by Multidimensional Scaling, Conjoint Triad Descriptors, and Sequence-Order-Coupling Number. Details of these features are described below. Except three features (CTD, SOCN, QSO), most of the features are not used in Ismail's work.

We extracted these features in this research:

i. Amino Acid Composition (AAC)

Using a protein sequence, we can calculate the fraction of each amino acid by implementing these feature descriptors [19]. This fraction is calculated using Equation 1, for all 20 amino acids:

$$fraction\ of\ aa_i = \frac{total\ of\ number\ of\ amino\ acid\ type\ i}{total\ number\ of\ amino\ acid\ in\ protein\ sequence} \qquad (1)$$

where a specific type of amino acid is symbolized by *i*.

ii. Dipeptide Composition (DPC)

Dipeptide Composition generates 400-dipeptide, fixed-length numerical information based on the input protein sequences. It measures the fraction of amino acids and their local order. It is calculated using Equation 2:

$$fraction\ of\ dep(i) = \frac{total\ of\ number\ of\ dep(i)}{total\ number\ of\ all\ posible\ dipeptide} \tag{2}$$

where *dep*(*i*) is one dipeptide *i* of 400 dipeptides.

iii. Normalized Moreau-Broto Autocorrelation Descriptors (NMB)

Before calculating Normalized Moreau-Broto Autocorrelation, we must define Moreau-Broto Autocorrelation. It can be defined using Equation 3:

$$AC(d) = \sum_{i=1}^{N-d} P_i P_{i+d} \tag{3}$$

where P_i and P_{i+d} are the amino acid properties at position *i* and *i*+*d*, respectively. Equation 4 is used to calculate Normalized Moreau-Broto Autocorrelation [20]:

$$ATS(d) = \frac{AC(d)}{(N-d)} \tag{4}$$

where *d*=1,2,3, ... ,30.

When we use PROFEAT, the value of nlag should be lower than the size of the sequence. Since the window size is 9, we set nlag=8.

iv. Moran Autocorrelation Descriptors (MORAN)

Moran Autocorrelation can be calculated using Equation 5:

$$I(d) = \frac{\frac{1}{N-d}\sum_{i=1}^{N-d}(P_i-\overline{P})(P_{i+d}-\overline{P})}{\frac{1}{N}\sum_{i=1}^{N}\left(P_i-\overline{P}\right)^2}\ d = 1,2,3,\ldots,30 \tag{5}$$

where $\overline{P}$ is the avarege of P_i. In the use of PROFEAT, we set nlag=8.

v. Geary Autocorrelation Descriptors (GEARY)

Geary Autocorrelation can be defined using Equation 6:

$$C(d) = \frac{\frac{1}{2(N-d)}\sum_{i=1}^{N-d}(P_i-P_{i+d})^2}{\frac{1}{N-1}\sum_{i=1}^{N}\left(P_i-\overline{P}\right)^2}\ d = 1,2,3,\ldots,30 \tag{6}$$

In the use of PROFEAT, we set nlag=8.

vi. Composition, Transition, Distribution (CTD)

These feature descriptors can be generated from protein sequences. It provides amino acid distribution patterns of a particular structural or physicochemical property [20] [21].

vii. Sequence-Order-Coupling Number (SOCN)

These feature descriptors are used to measure the amino acid distribution pattern of a specific physicochemical property along a protein sequence. The *d*th rank of sequence-order-coupling number can be calculated using Equation 7:

$$\tau_d = \sum_{i=1}^{N-d} (d_{i,i+d})^2 \;\; d = 1,2,3,\dots,30 \tag{7}$$

where $d_{i,i+d}$ is the distance between two amino acids at position *i* and $i+d$. In the use of protr, we also set nlag=8.

viii. Quasi-Sequence-Order Descriptors (QSO)

The QSO type-1 can be calculated using Equation 8:

$$X_r = \frac{f_r}{\sum_{r=1}^{20} f_r + w \sum_{d=1}^{30} \tau_d} \;\; r = 1,2,3,\dots,20 \tag{8}$$

where the normalized occurrence of amino acid type i is symbolized by f_r. In addition, w is the weighting factor, w=0.1. QSO type-2 is calculated using Equation 9.

$$X_d = \frac{w\tau_{d-20}}{\sum_{r=1}^{20} f_r + w \sum_{d=1}^{30} \tau_d} \;\; r = 21,22,23,\dots,50 \tag{9}$$

In the use of PROFEAT, we set nlag=8.

ix. Amphiphilic Pseudo-Amino Acid Composition (APAAC)

Before we calculate APAAC, we must define Pseudo-Amino Acid Composition (PAAC) [16]. Three original variables are generated, hydrophobicity values $H_1^0(i)$, hydrophilicity values $H_2^0(i)$, and side chain masses $M^0(i)$ of 20 amino acids (i=1,2,3, … ,20).

$$H_1(i) = \frac{H_1^0(i) - \sum_{i=1}^{20} \frac{H_1^0(i)}{20}}{\sqrt{\frac{\sum_i^{20} \left[H_1^0(i) - \sum_{i=1}^{20} \frac{H_1^0(i)}{20} \right]^2}{20}}} \tag{10}$$

$$H_2(i) = \frac{H_2^0(i) - \sum_{i=1}^{20} \frac{H_2^0(i)}{20}}{\sqrt{\frac{\sum_i^{20} \left[H_2^0(i) - \sum_{i=1}^{20} \frac{H_2^0(i)}{20} \right]^2}{20}}} \tag{11}$$

$$M(i) = \frac{M^0(i) - \sum_{i=1}^{20} \frac{M^0(i)}{20}}{\sqrt{\frac{\sum_i^{20}\left[M^0(i) - \sum_{i=1}^{20} \frac{M^0(i)}{20}\right]^2}{20}}} \tag{12}$$

Then, a correlation function can be generated as:

$$\theta(R_i, R_j) = \frac{1}{3}\left\{\left[H_1(R_i) - H_1(R_j)\right]^2 + \left[H_2(R_i) - H_2(R_j)\right]^2 + \left[M(R_i) - M(R_j)\right]^2\right\} \tag{13}$$

and sequence order-correlated factors can be calculated using Equation 14:

$$\theta_\lambda = \frac{1}{n-\lambda} \sum_{I=1}^{n-\lambda} \theta(R_i, R_{i+\lambda}), (\lambda < \mathrm{N}) \tag{14}$$

where λ is the parameter. The normalized frequency of 20 amino acids in the protein sequence is symbolized by f_i. A group of 20+λ feature descriptors, called the PAAC, can be calculated using Equation 15:

$$X_u = \frac{f_u}{\sum_{i=1}^{20} f_i + w \sum_{j=1}^{\lambda} \theta_\lambda}, when\ 1 \le u \le 20$$

$$X_u = \frac{w\theta_{u-20}}{\sum_{i=1}^{20} f_i + w \sum_{j=1}^{\lambda} \theta_\lambda}, when\ 20 + 1 \le u \le 20 + \lambda \tag{15}$$

where w=0.05. From Equation 10 and Equation 11, the hydrophobicity and hydrophilicity correlation can be defined as:

$$H_{i,j}^1 = H_1(i), H_1(j);\ \ H_{i,j}^2 = H_2(i), H_2(j) \tag{16}$$

Then, sequence order factor can be defined using Equation 17:

$$\tau_{2\lambda-1} = \frac{1}{N-\lambda} \sum_{i=1}^{N-\lambda} H_{i,i+\lambda}^1;\ \tau_{2\lambda} = \frac{1}{N-\lambda} \sum_{i=1}^{N-\lambda} H_{i,i+\lambda}^2,\ where\ \lambda < 2 \tag{17}$$

Finally, APAAC can be calculated using Equation 18:

$$p_u = \frac{f_u}{\sum_{i=1}^{20} f_i + \sum_{j=1}^{2\lambda} \tau_j}, when\ 1 \le u\ \le 20$$

$$p_u = \frac{w\tau_u}{\sum_{i=1}^{20} f_i + \sum_{j=1}^{2\lambda} \tau_j}, when\ 20 + 1 \le u\ \le 20 + \lambda \tag{18}$$

In the use of PROFEAT, we set the weight factor=0.05 and λ =8.

x. Total Amino Acid Properties (AAP)

Total Amino Acid Properties for a specific physicochemical property *i* is defined using Equation 19:

$$p_{tot(i)} = \frac{1}{N} \sum_{j=1}^{N} P_{norm_j^i} \tag{19}$$

where $P_{norm_j^i}$ represents the property *i* of amino acid R_j that is normalized between 0 and 1. *N* is the length of the protein sequence. $P_{norm_j^i}$ is calculated using Equation 20:

$$P_{norm_j^i} = \frac{(p_j^i - p_{min}^i)}{(p_{max}^i - p_{min}^i)} \tag{20}$$

where p_j^i is the original amino acid property *i* for the residue j. p_{max}^i and p_{min}^i are the maximum and the minimum values of the original amino acid property i, respectively.

xi. Position Specific Scoring Matrix (PSSM)

PSSM features were generated using PSI-BLAST against a local database generated from the phosphorylation data set. For each protein sequence (window size 9), PSI-BLAST creates matrix (9× 20 amino acid). We then create a 180-length vector for each sequence.

xii. BLOSUM and PAM Matrices for the 20 Amino Acid (BLOSUM)

These descriptors are generated from BLOSUM and PAM. In the use of protr, we set k=5, lag=3, and Matrix type=AABLOSUM45.

xiii. Amino Acid Properties Based Scales Descriptors (Protein Fingerprint) (ProtFP)

These descriptors are scaled-based generated from AAIndex properties. In the use of protr, we set pc=5, lag=5, index vector for Amino Acid Index =(160:165, 258:296).

xiv. Scales-based Descriptor derived by Principal Components Analysis (SCALES)

These descriptors are generated using principal components analysis. In the use of protr, we set pc=7, lag=5, properties matrix=AAindex (7:26).

xv.Scales-based Descriptor derived by Multidimensional Scaling (MDDSCALES)

Scales-based Descriptors are derived by Multidimensional Scaling. These descriptors are calculated by using multidimensional scaling. In the use of protr, we set lag=8.

BLOSUM, PROTFP, SCALES, and MDDSCALES descriptors are often implemented in Proteochemometric Modeling (PCM).

xvi. Conjoint Triad Descriptors (CTriad)

Introduced by Shen et al. [22], these descriptors provide information about paired base protein based on amino acid classification. Every protein sequence is represented by a numerical vector space containing amino acid descriptors. Several groups were created to cluster the 20 kinds of amino acid, based on information of dipoles and the volumes of their side chains. There are two steps to create these descriptors. First, the amino acid is classified into seven groups based on the dipole scale and volume scale. The next step is to calculate the conjoint triad. There are three points for calculation: the properties of an amino acid, its surrounding amino acids, and the consideration of three continuous amino acids as one unit.

3.2.3 Protein feature selection using Random Forest

Features used in this research were generated from 3 different tools that generate 16 feature descriptors. We implemented Random Forest for feature selection. We listed the important features based on the Gini Impurity index.

3.2.4 Support Vector Machine for phosphorylation site prediction

To classify whether a residue is phosphorylated, we used Support Vector Machine. We implemented Gaussian as the kernel.

3.2.5 Evaluation

i. Evaluation metrics

We conducted an evaluation to measure and compare the performance of classification results. Table 3.4 shows the combination of results of prediction compared to the results of real observations. True positive (TP) and True Negative (TN) occur when the result of the prediction is the same as the outcome of the real observation. False Positive (FP) and False Negative (FN) occur when the result of the prediction is different from the outcome of real observation.

Table 3.4 Combination of prediction outcomes with observation matrix

		Predicted Condition	
		Positive	Negative
True Condition	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Using Table 3.4, we analyzed and compared the classification based on these metrics:

Accuracy

Accuracy is a measurement to calculate the proportion of the number of times the classification predicted the result correctly. We computed accuracy using Equation 21:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (21)$$

Sensitivity

We use sensitivity to measure the proportion of the actual positive result which is classified correctly. By using Equation 22 we could compute the sensitivity value.

$$Sensitiviy = \frac{TP}{TP+FN} \qquad (22)$$

Specificity

Specificity is a measurement that calculates the classification performance of predicting negative results correctly. We computed the value of specificity using Equation 23:

$$Specificity = \frac{TN}{TN+FP} \tag{23}$$

F1 score

F1 score is another type of accuracy measurement. It evaluates the proportion of precision and recall in the classification result. The F1 score could be measured using Equation 24:

$$F1\ score = 2 \times \frac{TP}{TP+FP+FN} \tag{24}$$

Matthews Correlation Coefficient (MCC)

MCC was introduced by Matthews, B.W in 1975. It is commonly used to measure the performance of a binary classification [23]. The value of MCC could be obtained using Equation 25.

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FP) \times (TN+FN)}} \tag{25}$$

Receiver Operating Characteristic (ROC) Curve

An ROC curve is a commonly used way to visualize and evaluate the performance of a binary classifier. ROC compares the values of True Positive Rate with the False Positive Rate.

3.2.6 Grid search

Grid search is a method of finding the best number of features that achieve the highest accuracy for classification. This method consisted of two phases.

In the first phase, we defined the class label and the features. Then we split the data set into two sets, a data for training and a data for testing, by using *k*–fold cross validation. This is illustrated in Figure 3.2. Using the training data, we created a model with Random Forest and listed the important features. We then set the grid length (for example, grid length=20), selected the number of features, and added numbers of features based on grid length. Using the selected number of features, we conducted cross validation for each number of feature selection. We selected the best number of features (X) that produced the highest accuracy from cross validation.

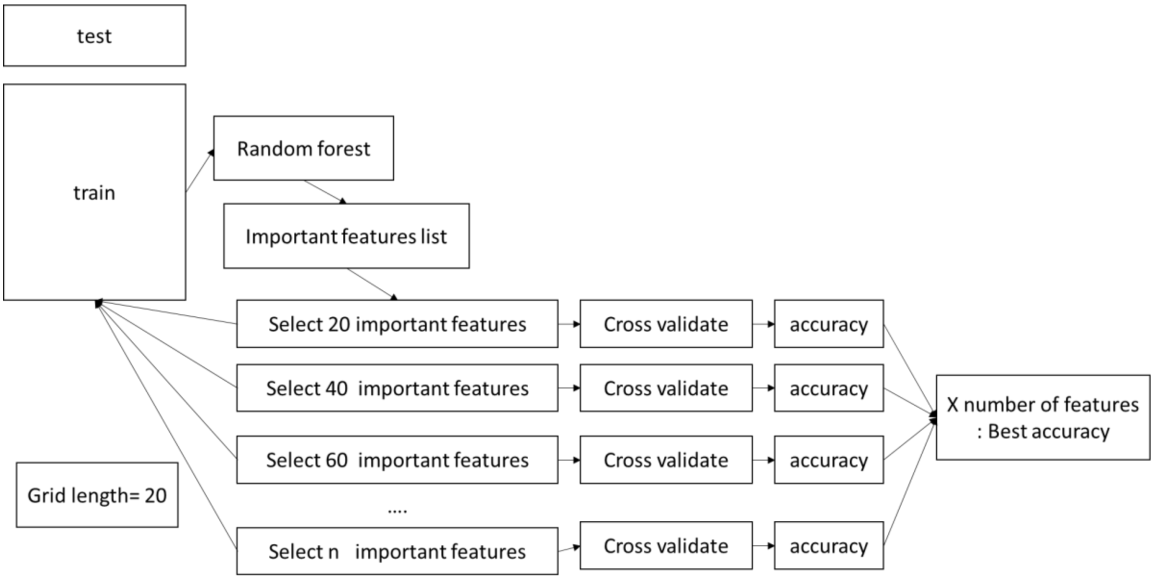


Figure 3.2 First phase in grid search

In phase two, we conducted a finer grid search than phase one as shown in Figure 3.3. The feature numbers that were selected were based on the numbers within the grid length of X. By selecting those feature numbers, we conducted cross validation. We then selected the number of the feature that had the highest accuracy (Y). Using the important list, we then selected Y number of features for the test and training data. We then generated a new model from the selected features in the training data and tested the model using the test data set, in which we also selected Y number of features. We conducted grid search for each fold. In addition, we recorded the result of the prediction.

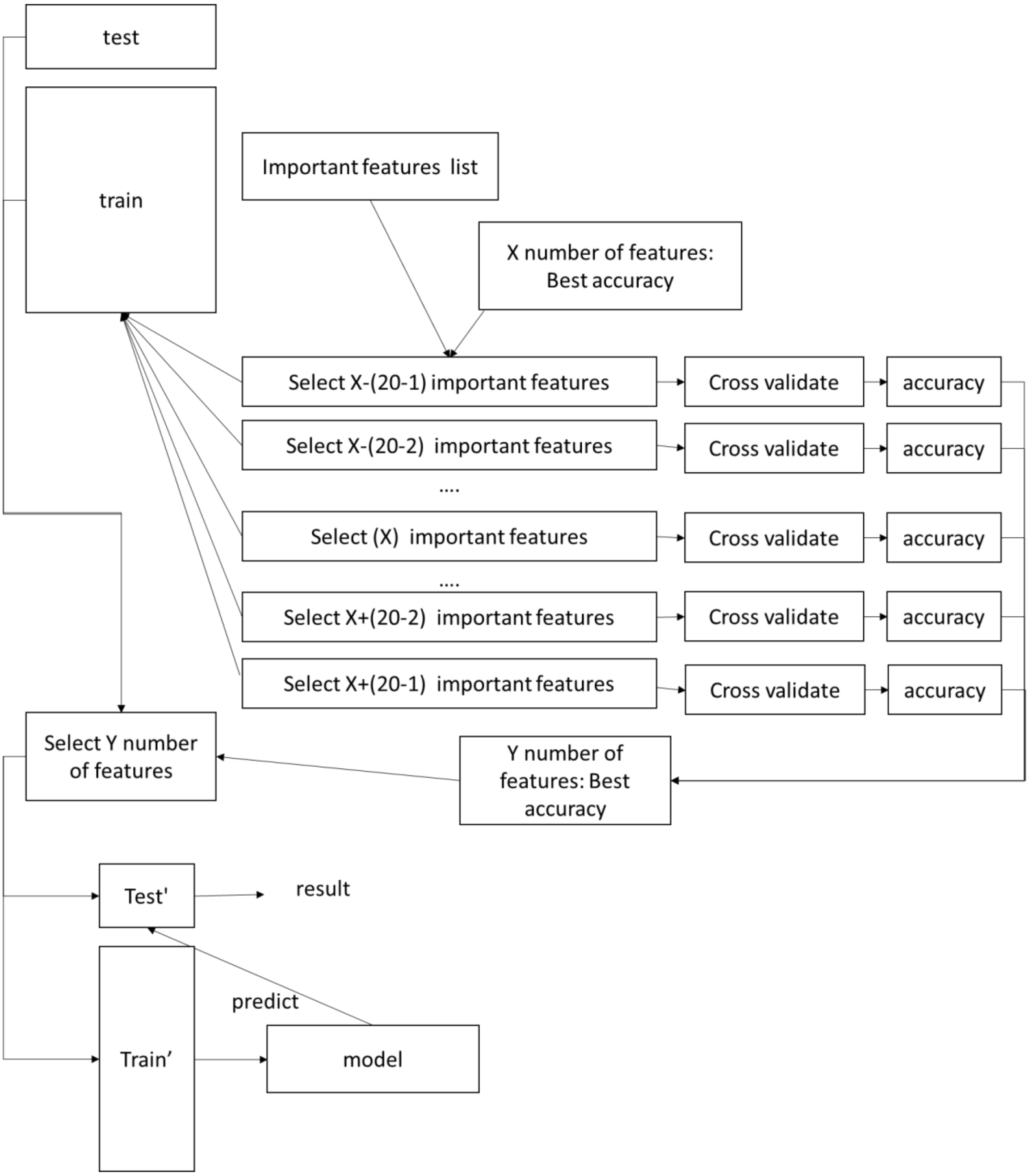

test
train
Important features list
X number of features: Best accuracy
Select X-(20-1) important features
Select X-(20-2) important features
....
Select (X) important features
....
Select X+(20-2) important features
Select X+(20-1) important features
Cross validate
accuracy
Select Y number of features
Y number of features: Best accuracy
Test'
result
predict
Train'
model

Figure 3.3 Second phase in grid search

The main goal of using grid search was to decrease the time of computation. If we were to use brute-force comparison to find the best number of features from the data set containing *f* number of features, it would require *f* comparison processes. However, using the grid search method, we can lower the computational time into *(f/grid length)+2(grid length)* number of processes, where the value *f* is much larger than *grid length.*

Chapter 4 Result and discussion

This chapter will explain the result of feature selection. The classification result for phosphorylation site prediction using the two data sets (P.ELM and PPA) will also be explained. We will also compare our classification result and features with the results from previous work related to our topic.

4.1 P.ELM data set

4.1.1 Important features

We conducted classification using the P.ELM data set. To evaluate the performance, we used ten times 10-fold cross validation. For each fold in each iteration, the model generates a list of important features measured using Gini Impurity Index (GII). Thus, there were 100 lists of important features. We averaged the GII value of each feature in the 100 lists and conducted a detailed analysis to determine which features were dominant and most influenced the classification method. In addition, we averaged the numbers of features in the 100 lists to show how many features are selected in average (see Table 4.2 below).

Figure 4.1 shows the list of features and their average GII values for Serine residue. The top three important values ranged from 38.26 to 47.10. In addition, as shown in the chart, there were only a few features of Serine that showed a significant importance. The important features in this data set are Amino Acid Composition (AAC) and Quasi-Sequence-Order Descriptors (QSO) which occupy the top three highest GII values.

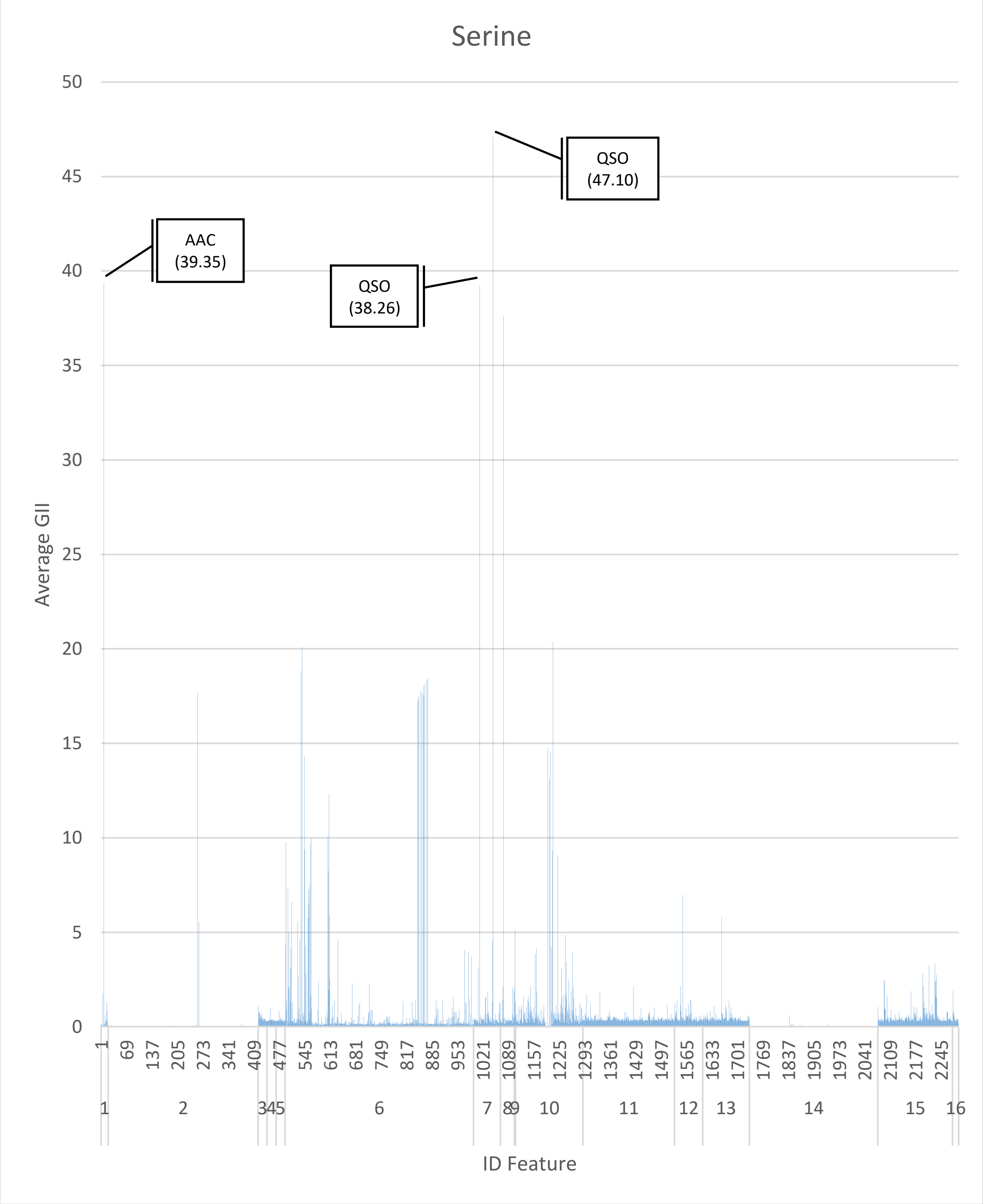


Figure 4.1 Distribution of important features of Serine from P.ELM data set. The numbers 1-16 attached under the feature numbers indicate the 16 different feature groups (e.g. AAC and QSO).

The highest value of the top important features in Threonine is lower than Serine. Figure 4.2 shows the top three values of important features, which are between 11.51 to 12.04. These important features are Amino Acid Composition (AAC) and Amphiphilic Pseudo-Amino Acid Composition (APAAC).

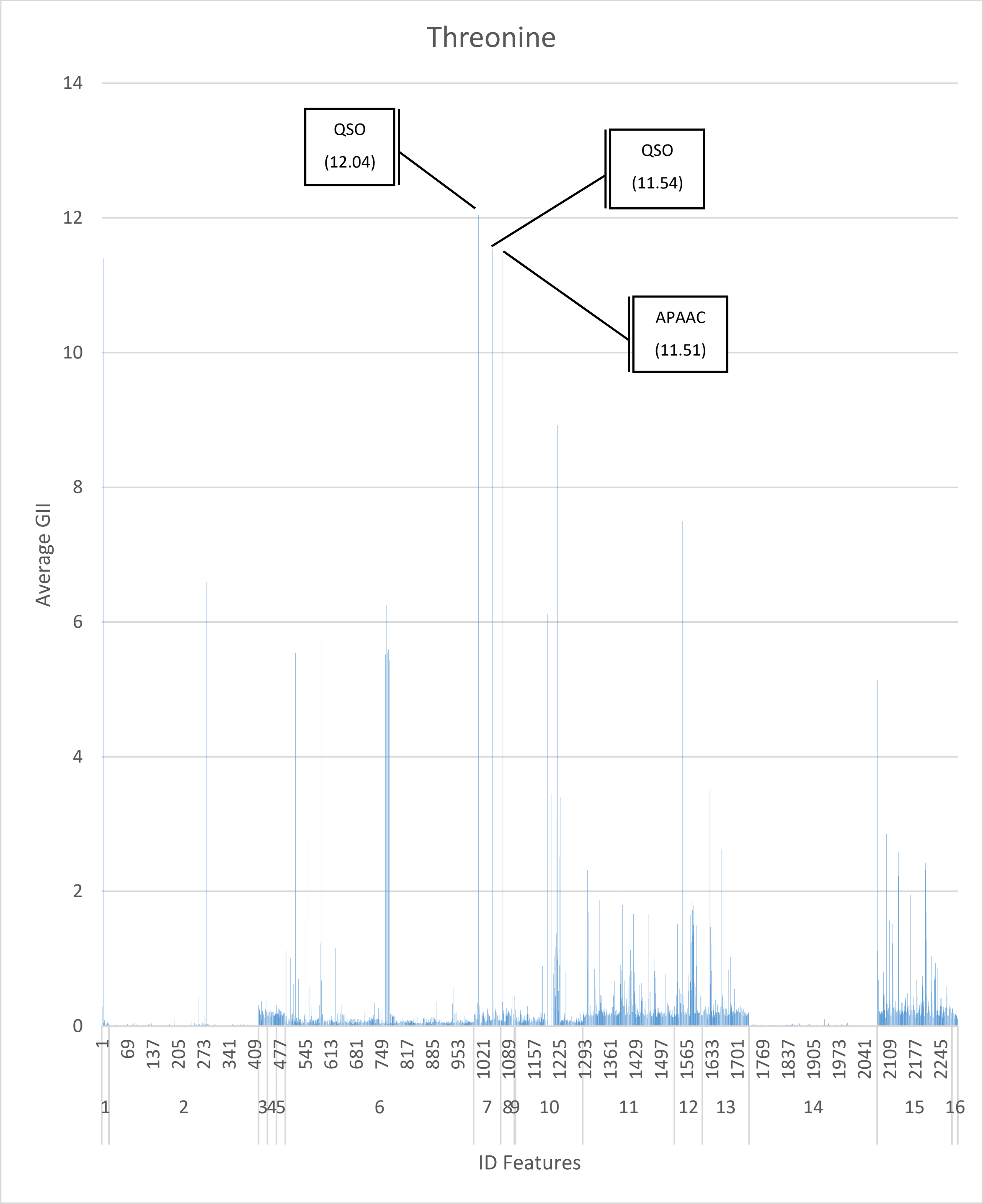


Figure 4.2 Distribution of important features of Threonine from P.ELM data set. The numbers 1-16 attached under the feature numbers indicate the 16 different feature groups (e.g. QSO and APAAC).

The third residue in the P.ELM data set is Tyrosine. The average GII value of the top three important features is lower than Serine and Tyrosine. The GII values of the top three important features are between 1.37 to 1.76 as shown in Figure 4.3. Composition, Transition, Distribution

(CTD), Quasi-Sequence-Order Descriptors (QSO), and Amphiphilic Pseudo-Amino Acid Composition (APAAC) are the top three important features in the Tyrosine data set.

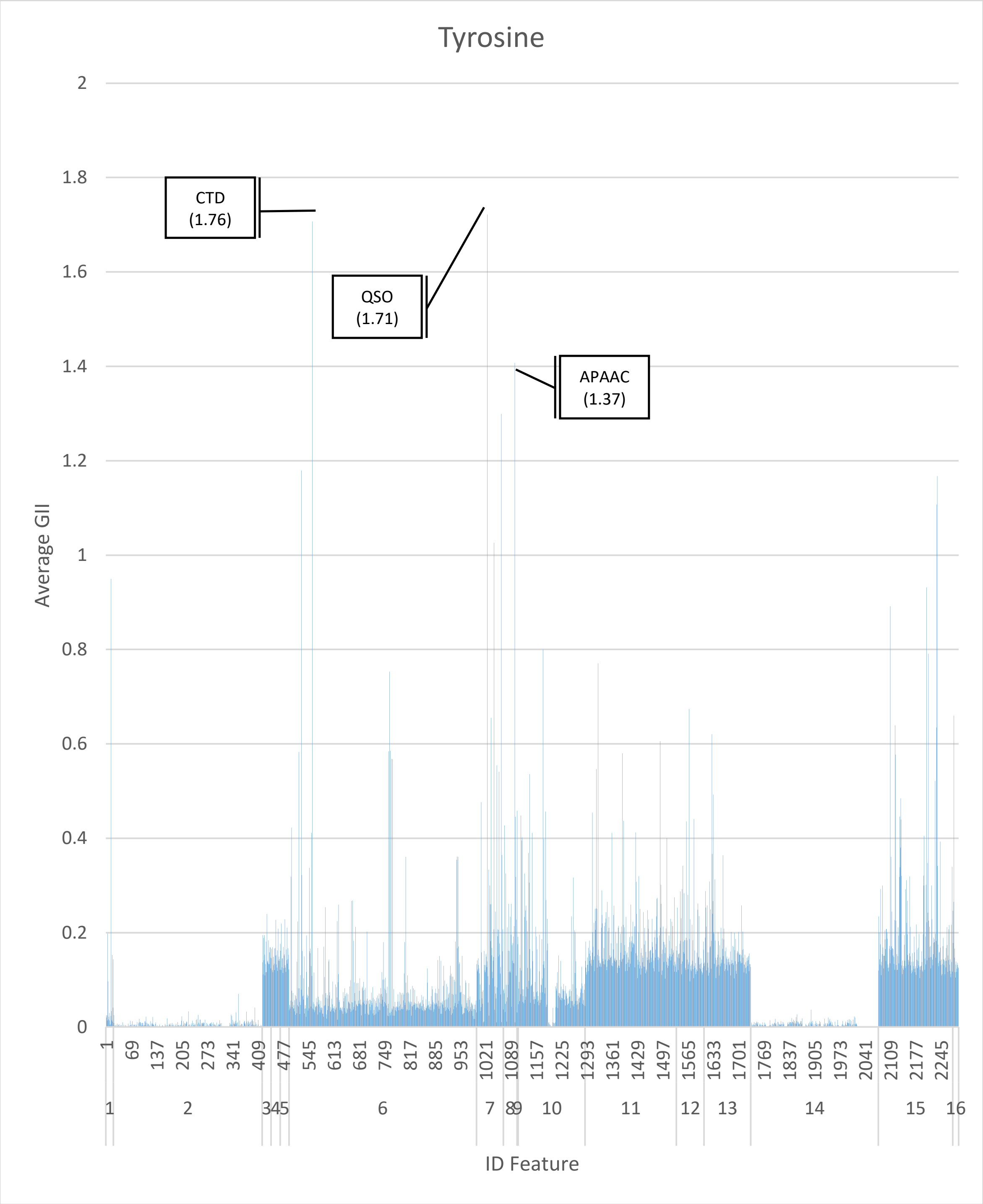


Figure 4.3 Distribution of important features of Tyrosine from P.ELM data set. The numbers 1-16 attached under the feature numbers indicate the 16 different feature groups (e.g. CTD, QSO, and APAAC).

An important features comparison was conducted for the P.ELM data set. We listed the top 20 important features for each residue, as shown in Table 4.1. For Serine, the Composition, Transition, and Distribution (CTD) feature is the most prevalent important feature. We can see the same result for Threonine. However, for Tyrosine, the Scales-based Descriptor derived by Multidimensional Scaling (MDSSCALES) feature is the most prevalent important feature.

Table 4.1 List of top 20 important features in the P.ELM data set for Serine, Threonine, and Tyrosine residues

Rank	Serine	Threonine	Tyrosine
1	QSO	QSO	QSO
2	AAC	QSO	CTD
3	QSO	APAAC	APAAC
4	APAAC	AAC	QSO
5	PSSM	PSSM	CTD
6	CTD	BLOSUM	MDSSCALES
7	CTD	DPC	MDSSCALES
8	CTD	CTD	QSO
9	CTD	PSSM	AAC
10	CTD	SCALES	MDSSCALES
11	CTD	CTD	MDSSCALES
12	CTD	CTD	PSSM
13	DPC	CTD	MDSSCALES
14	CTD	CTD	SCALES
15	CTD	CTD	CTD
16	CTD	CTD	BLOSUM
17	CTD	MDSSCALES	SOCN
18	CTD	PROTFP	QSO
19	PSSM	PSSM	MDSSCALES
20	PSSM	PSSM	MDSSCALES

4.1.2 Classification result

In this research, we conducted a detailed analysis of feature selection for each residue data set. The metrics we used for measuring the performance were Accuracy, Area Under ROC Curve (AUC), Sensitivity, Septicity, F1 Score, and Matthews Correlation Coefficient (MCC). From the result of the 10-fold cross validation conducted 10 times, we measured the average of each evaluation metric.

By implementing feature selection with grid search for finding the best set of features, performances were greatly improved, as shown in Table 4.2. For instance, Serine increased its accuracy and had the highest accuracy at 96.46% using 373.45 important features in average (i.e. the average number of features selected in 10 times 10-fold cross validation). This is followed by Threonine at 91.75% using its averaged 296.71 important features. Tyrosine achieved its best performance, 76.77%, using its averaged 402.69 important features. Based on the comparison of before and after using feature selection, Threonine had the largest percentage of increase in accuracy, 26.08%, followed by Serine, 24.68%, and Tyrosine, 12.44%.

Since feature selection decreased the performance in Ismail's work, it is an important finding in this study that under an appropriate combination of classifier and features, feature selection could improve the performance of protein phosphorylation site prediction.

Table 4.2 Performance of classification using all of the features (2292 features) and best result of features selection for P.ELM data set

Metrics	Serine		Threonine		Tyrosine	
	All features	Average 373.45 features	All features	Average 296.71 features	All features	Average 402.69 features
Accuracy	0.7174	0.9642	0.6567	0.9175	0.6433	0.7677
AUC	0.7171	0.9642	0.6567	0.9168	0.6387	0.7639
Sensitivity	0.7946	0.9701	0.8581	0.9197	0.6968	0.8097
Specificity	0.6396	0.9582	0.3425	0.9139	0.5805	0.7181
F1 Score	0.7382	0.9645	0.7526	0.9314	0.6783	0.7906
MCC	0.4404	0.9285	0.2381	0.8282	0.2814	0.5309

4.2 PPA data set

4.2.1 Important features

For the PPA data set, we also conducted classification. We evaluated performance using Leave-One-Out cross validation. Based on each fold, using Random Forest, an important feature list was generated from the training data. Therefore, the number of important feature lists generated equals the number of observations in the data set. As in the P.ELM data set, we measured the average value of each feature importance and the number of features in all the feature lists.

Figure 4.4 shows the list of features and their average GII values for Serine residue. The top three important values ranged from 5.48 to 6.49. In addition, as shown in the chart, there were only a few features that showed a significant importance also for Serine. The important features in this data set are Amphiphilic Pseudo-Amino Acid Composition (APAAC) and Quasi-Sequence-Order Descriptors (QSO) which occupy the top three highest GII values.

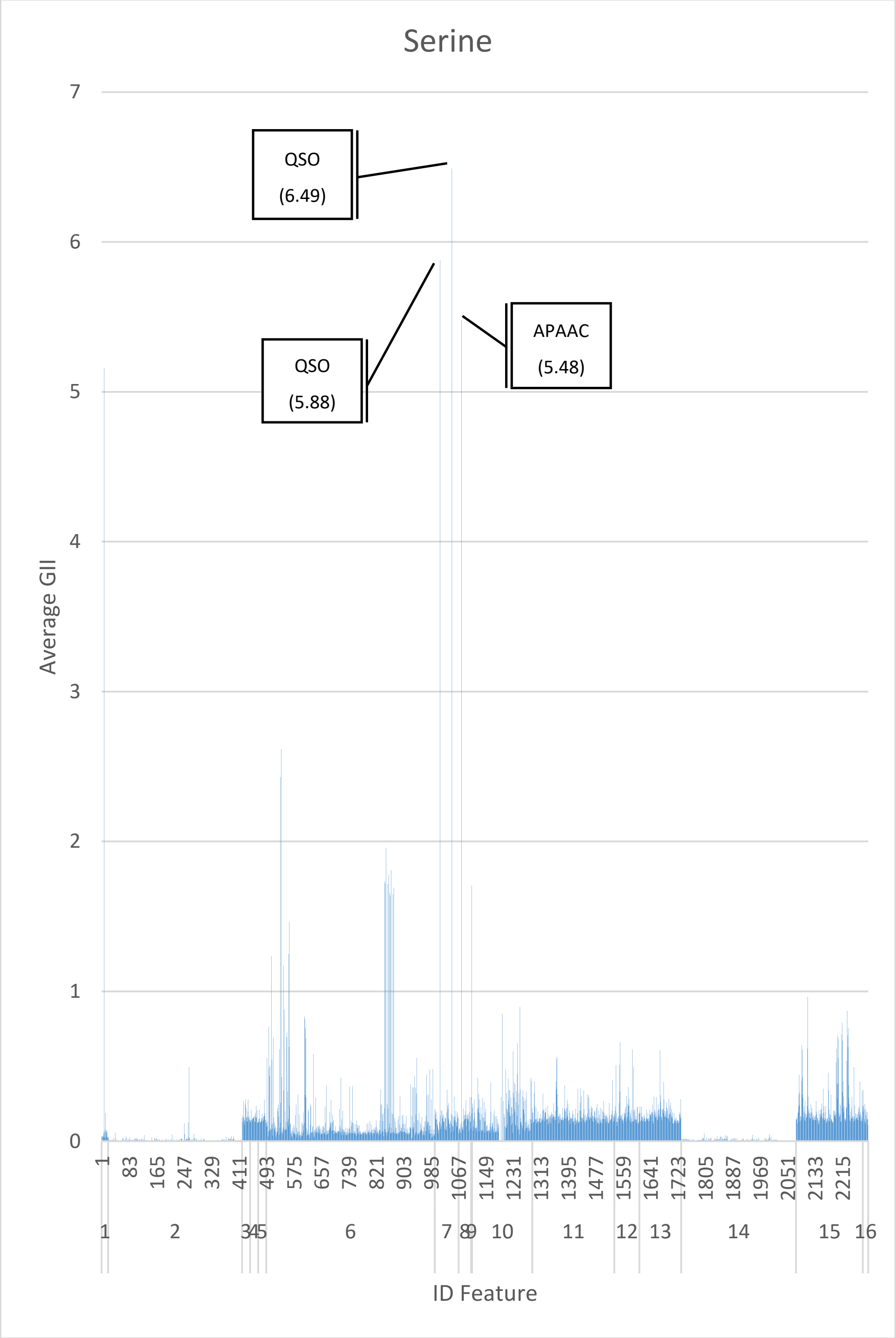


Figure 4.4 Distribution of important features of Serine from PPA data set. The numbers 1-16 attached under the feature numbers indicate the 16 different feature groups (e.g. QSO and APAAC).

The highest value of the top important features in Threonine is lower than Serine. Figure 4.5 shows the top three values of important features ranging from 1.51 to 1.61. These important features are Amphiphilic Pseudo-Amino Acid Composition (APAAC) and Quasi-Sequence-Order Descriptors (QSO).

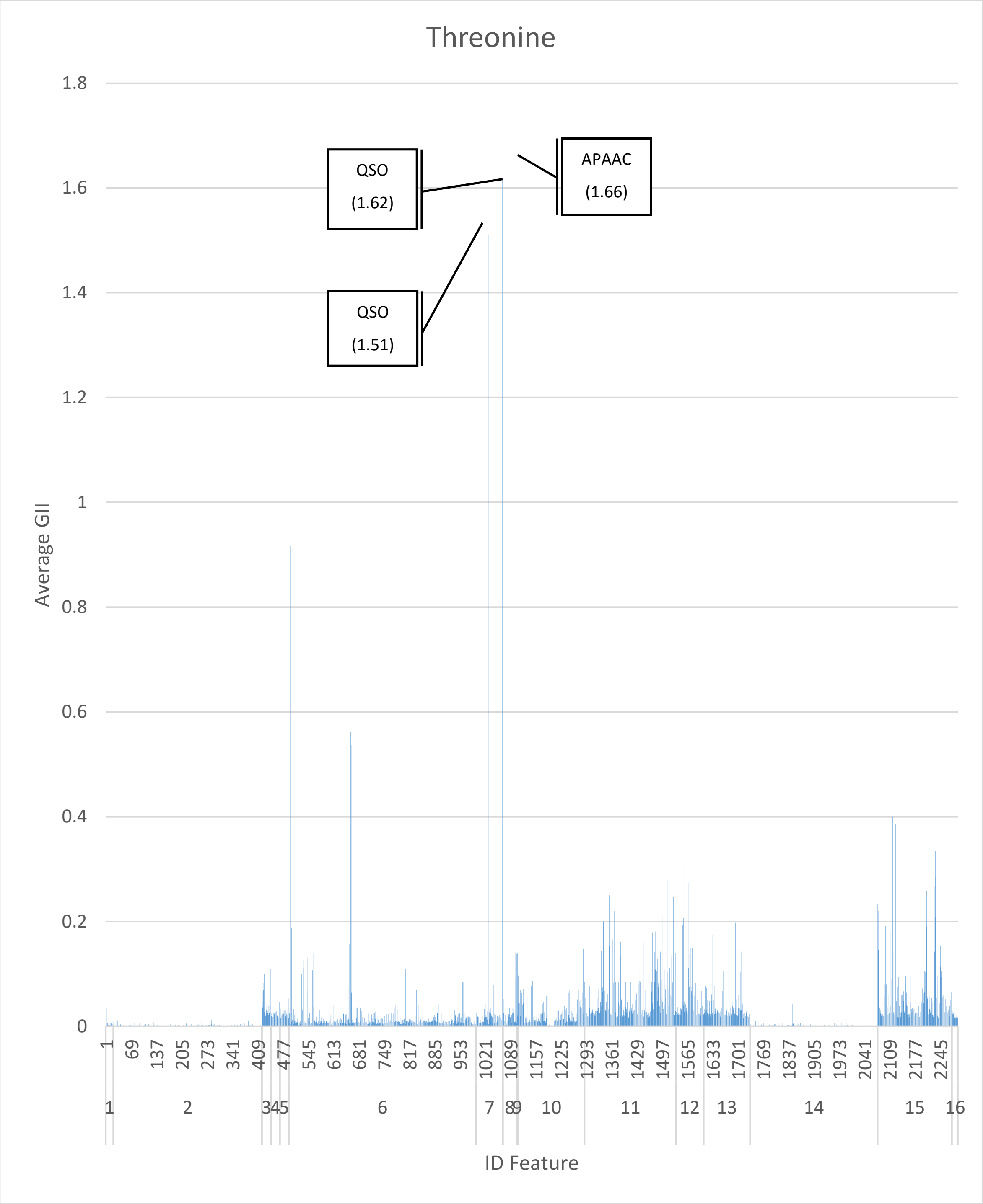


Figure 4.5 Distribution of important features of Threonine from PPA data set. The numbers 1-16 attached under the feature numbers indicate the 16 different feature groups (e.g. APAAC and QSO).

The third residue in the PPA data set is Tyrosine. The average GII value of the top three important features is lower than Serine and Tyrosine. The GII values of the top three important features are between 0.34 to 0.35 as shown in. Figure 4.6. Quasi-Sequence-Order Descriptors (QSO), Total Amino Acid Properties (AAP), and Sequence-Order-Coupling Number (SOCN) are the top three important features in the Tyrosine data set.

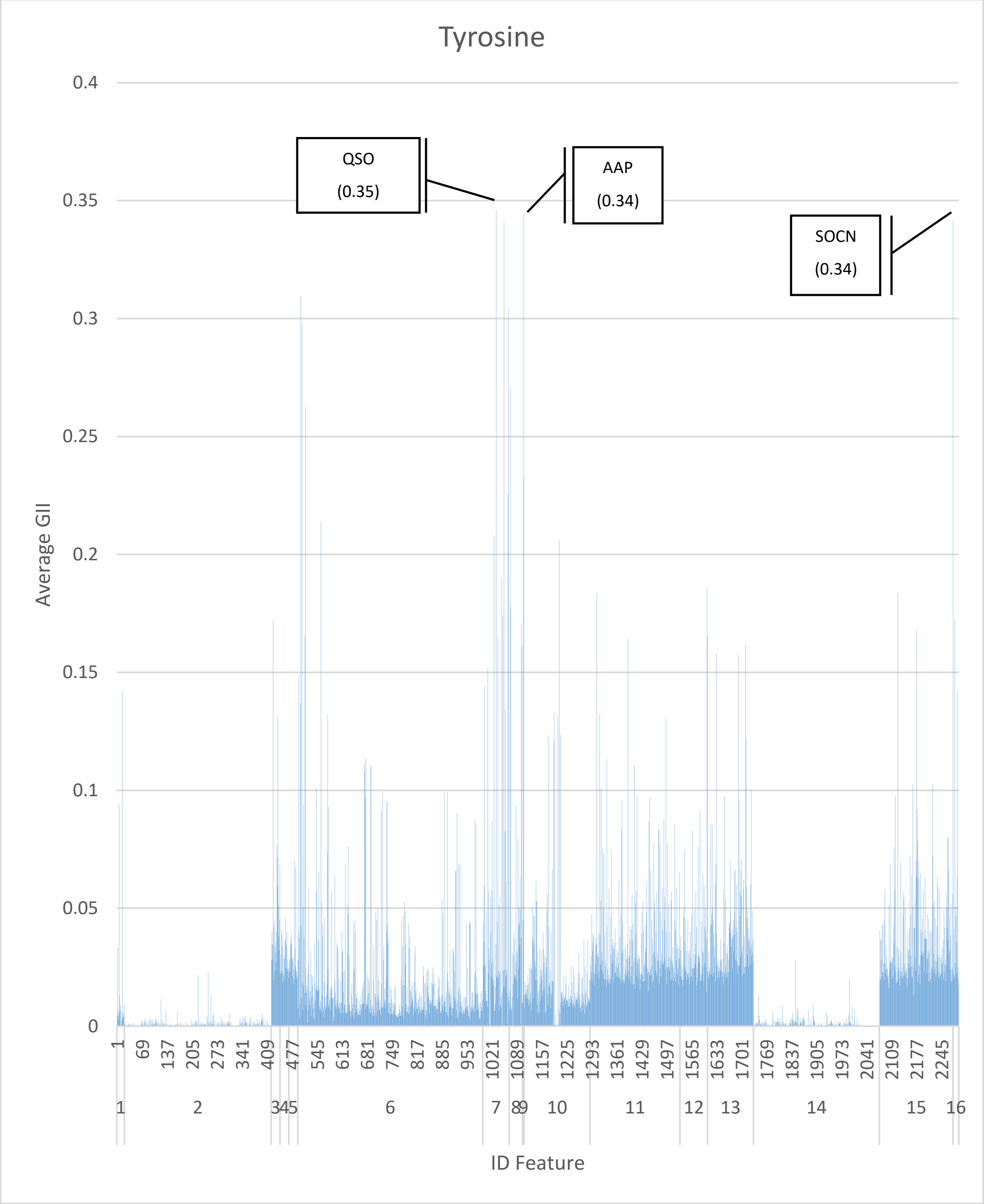


Figure 4.6 Distribution of important features of Tyrosine from PPA data set. The numbers 1-16 attached under the feature numbers indicate the 16 different feature groups (e.g. QSO, AAP, and SOCN).

Important feature comparison is also conducted for the PPA data set. We list top 20 important feature for each residue as shown in Table 4.3 List of top 20 important features in the PPA data set for Serine, Threonine, and Tyrosine residues. For Serine, Composition, Transition, Distribution (CTD) feature is the most prevalent important feature. For Threonine, Scales-based Descriptor derived by Multidimensional Scaling (MDSSCALES) feature is the most prevalent important feature. Finally, for Tyrosine, Quasi-Sequence-Order Descriptors (QSO), feature is the most prevalent important feature.

Table 4.3 List of top 20 important features in the PPA data set for Serine, Threonine, and Tyrosine residues

Rank	Serine	Threonine	Tyrosine
1	QSO	APAAC	QSO
2	QSO	QSO	AAP
3	APAAC	QSO	SOCN
4	AAC	AAC	QSO
5	CTD	CTD	CTD
6	CTD	CTD	QSO
7	CTD	APAAC	CTD
8	CTD	QSO	APAAC
9	CTD	QSO	CTD
10	CTD	AAC	AAP
11	CTD	CTD	QSO
12	CTD	CTD	CTD
13	AAP	MDSSCALES	QSO
14	CTD	MDSSCALES	PSSM
15	CTD	MDSSCALES	QSO
16	CTD	MDSSCALES	BLOSUM
17	CTD	BLOSUM	MDSSCALES
18	CTD	MDSSCALES	SCALES
19	CTD	SCALES	APAAC
20	CTD	MDSSCALES	QSO

4.2.2 Classification result

In general, as shown in Table 4.4, we can see that without feature selection the accuracy is lower than 70% for all three data sets. However, there is an improvement if we implement feature selection before conducting class prediction. Threonine has the highest accuracy, 86.76%, using the averaged 521.49 important features. This is followed by Serine, achieving 84.73% accuracy using the averaged 403.98 important features. Tyrosine has the lowest accuracy, achieving 77.45% using the averaged 264.18 important features.

If we compare the increase in performance between not using feature selection and feature selection, Threonine achieved a 30.88% increase in accuracy, followed by Serine's 27.30% increase. Tyrosine has the lowest increase of accuracy at 10.78%.

Table 4.4 Performance of classification using all of the features (2292 features) and best result of features selection for PPA data set

Metrics	Serine		Threonine		Tyrosine	
	All features	Average 403.98 features	All features	Average 521.49 feature	All features	Average 264.18 feature
Accuracy	0.5863	0.8593	0.5588	0.8676	0.6667	0.7745
AUC	0.5863	0.8593	0.5588	0.8676	0.6667	0.7745
Sensitivity	0.7687	0.8586	0.4412	0.8529	0.6471	0.7647
Specificity	0.4039	0.8599	0.6765	0.8823	0.6863	0.7843
F1 Score	0.6502	0.8592	0.5	0.8657	0.66	0.6531
MCC	0.1854	0.7186	0.1210	0.7356	0.3336	0.5491

4.3 Comparison with other previous works

In this research, we compared the result from our method with several other previous research works on phosphorylation site prediction as shown in Table 4.5. The compared methods are as follows: Netphos [24] , NetphosK [3], GPS 2.1 [6], Swaminathan, PPRED [25], Musite [26], PhosphoSVM [7], and RF-Phos [8]. Most of the previous research did not conduct feature selection to improve the classification of phosphorylation sites. Only RF-Phos implemented feature selection using Random Forest.

Table 4.5. List of related phoshphorylation site prediciton research

Method	Researchers	Year	Feature Selection	Classifier
NetPhosK	Blom et al.	2004	-	Neural Network
GPS 2.1	Xue et al.	2011	-	Motif Length Selection
Swaminathan	Swaminathan et al.	2010	-	Epsilon-SVR
Netphos	Blom et al.	1999	-	Neural Network
PPRED	Biswas et al.	2010	-	SVM
Musite	Gao et al.	2010	-	KNN
PhoshoSVM	Dou et al.	2014	-	SVM

RF-Phos	Ismail et al.	2016	Random Forest	Random Forest

4.3.1 Classification result

P.ELM Data Set

In this work, we also compared the result from the P.ELM data set and the PPA data set with other results from previous research. Table 4.6 shows the performance comparison between our results and other results. For Serine and Threonine, our method achieved the highest AUC, sensitivity, and MCC values. However, our specificity value from the Threonine data set is lower than the result of RF-Phos. On the other hand, in the Tyrosine data set our method achieved a lower AUC, specificity, and MCC, in comparison with the result of RF-Phos.

Table 4.6 Performance comparison of several phosphorylation site prediction methods for Serine, Threonine, and Tyrosine residues using the P.ELM data set

Methods	Serine				Threonine				Tyrosine			
	AUC	Sen	Spec	MCC	AUC	Sen	Spec	MCC	AUC	Sen	Spec	MCC
NetPhosK	0.63	0.509	0.678	0.08	0.60	0.620	0.568	0.07	0.60	0.395	0.742	0.08
GPS 2.1	0.73	0.331	0.933	0.20	0.70	0.381	0.923	0.20	0.61	0.345	0.789	0.08
Swaminathan	0.70	0.313	0.887	0.13	0.72	0.280	0.925	0.14	0.62	0.605	0.570	0.09
NetPhos	0.70	0.341	0.867	0.12	0.66	0.343	0.837	0.09	0.65	0.347	0.845	0.13
PPRED	0.75	0.323	0.916	0.17	0.73	0.303	0.910	0.13	0.70	0.430	0.827	0.17
Musite	0.81	0.414	0.937	0.25	0.78	0.338	0.948	0.22	0.72	0.384	0.867	0.18
PhosphoSVM	0.84	0.444	0.940	0.30	0.82	0.378	0.950	0.25	0.74	0.419	0.873	0.21
RF-Phos	0.88	0.840	0.850	0.65	0.90	0.830	**0.940**	0.70	**0.91**	**0.830**	**0.880**	**0.70**
Our Method	**0.96**	**0.970**	**0.958**	**0.93**	**0.92**	**0.920**	0.914	**0.83**	0.77	0.810	0.759	0.53

PPA Data Set

We also compared our classification results with the results in other research. The methods we compared are: NetphosK, GPS 2.1, NetPhos, PHOSPHER, Musite, PhosphoSVM, and RF-Phos. In Table 4.7, we can see that our method has a lower performance in sensitivity and specificity, for all residues. However, achieving the best MCC for all residues is of higher importance.

Table 4.7 Performance comparison of several phosphorylation site prediction methods for Serine, Threonine, and Tyrosine residues using the PPA data set

Methods	Serine			Threonine			Tyrosine		
	Sen	Spec	MCC	Sen	Spec	MCC	Sen	Spec	MCC
NetPhosK	0.8013	0.3879	0.10	0.6912	0.5082	0.06	0.2549	0.8323	0.04
GPS 2.1	**0.9479**	0.2862	0.14	**0.9559**	0.2084	0.07	0.9804	0.2142	0.09
NetPhos	0.7655	0.5420	0.16	0.5441	0.7743	0.12	0.6471	0.6750	0.13
PHOSFER	0.7459	0.6551	0.22	0.7794	0.6477	0.14	0.6275	0.5929	0.08
Musite	0.5570	**0.8739**	0.31	0.4853	**0.9355**	0.26	0.4706	**0.8877**	0.20
PhosphoSVM	0.6384	0.8176	0.29	0.7059	0.8176	0.19	**0.8235**	0.6418	0.18
RF-Phos	0.7200	0.7000	0.41	0.7900	0.7000	0.50	0.6100	0.6200	0.29
Our Method	0.8430	0.8556	**0.68**	0.8529	0.8824	**0.74**	0.7647	0.7843	**0.55**

4.3.2 Feature selection

Table 4.8 shows a comparison of the top ten important features used in our method and RF-Phos. Both of these lists are used to classify phosphorylation sites using the P.ELM data set. In the RF-Phos list, CTD is the most prevalent feature, followed by QSO. The number one rank for each residue is also occupied by CTD and QSO.

In our method, the CTD and QSO features are also important features. However, there are new additional features, including: AAC, APAAC, PSSM, BLSOSUM, DPC, and SCALES. For the Serine and Threonine residues, the new features improve performance. However, for Tyrosine, there was no improvement in classification performance. This lack of improvement may have been caused by the addition of new features or the absence of several features from previous methods.

Table 4.8 Comparison of the top 10 important features between RF-Phos and our method for phosphorylation site prediction using the P.ELM data set

Rank	RF-Phos			Our Method		
	Serine	Threonine	Tyrosine	Serine	Threonine	Tyrosine
1	QSO	QSO	CTD	QSO	QSO	QSO
2	OP	QSO	CTD	AAC	QSO	CTD
3	QSO	SF	ASA	QSO	APAAC	APAAC
4	SF	OP	IG	APAAC	AAC	QSO
5	CTD	CTD	OP	PSSM	PSSM	CTD
6	ACH	CTD	CTD	CTD	BLOSUM	MDSSCALES
7	ACH	CTD	CTD	CTD	DPC	MDSSCALES
8	ASA	OP	CTD	CTD	CTD	QSO
9	CTD	CTD	CTD	CTD	PSSM	AAC
10	ASA	CTD	ASA	CTD	SCALES	MDSSCALES

Chapter 5 Summary and future work

We conclude our thesis by explaining the summary of accomplished work. We also suggest ideas and topics for future research to improve results.

5.1 Summary

One of the most common types of post-translational modification in the eukaryotic cell is phosphorylation. This occurs when a phosphate group attaches to a residue in the protein sequence. Phosphorylation commonly occurs at the Serine, Threonine, or Tyrosine residues. It is also important for cellular activities, such as cell growth and intracellular signal transduction. Many research works have been conducted to predict phosphorylation sites using the experimental and computational approaches. The computational approach, in particular the non-kinase-specific approach, is being studied intensively in recent years. This is because of improvements in computer technology and the advancement of machine learning algorithms.

In this research, we conducted predictions for phosphorylation sites using the non-kinase-specific approach. We used the P.ELM data set which consists of phosphorylation sites from humans and several species of animal. In addition, we used the PPA data set as a small independent data set, which consists of plant phosphorylation site information. Random Forest was implemented for feature selection. We listed the important features using Gini Impurity Index. By implementing grid search we found the numbers of features that achieved the highest classification performance for each residue. We classified the phosphorylation sites by using Support Vector Machine.

In this study using the P.ELM data set, we (i) outperformed the classification performance from previous research for the Serine and Threonine data sets. However, the classification performance using Tyrosine data could not be improved. For PPA data set, our method achieved the highest MCC value for all residues.

(ii) Feature selection was implemented in previous research. However, the classification performance decreased. Conversely, by implementing feature selection in our method, we could increase the performance of phosphorylation site classification. We conducted a grid search to find the best number of features to increase the classification performance.

(iii) We introduced new features to improve Phosphorylation site classification. These features are Amino Acid Composition (AAC), Amphiphilic Pseudo-Amino Acid Composition (APAAC), and Position Specific Scoring Matrix (PSSM). Our method also implemented features from previous works, which are Composition, Transition, Distribution Descriptors (CTD), and Quasi-Sequence-Order Descriptor (QSO).

5.2 Future work

In this study, we proposed new features to be implemented for the classification of phosphorylation sites. These new features consisted of numerical information representing the physicochemical properties of each amino acid in the protein sequence.

We hope future work can discover new features that may improve classification performance. Feature selection in this thesis is conducted using three tools PROFEAT, PSIBlast, and protr to generate 16 different feature descriptors. We suggest finding new features, not only numerical but also categorical, which can increase the performance of phosphorylation site prediction.

Future research should explore new combinations of new features with features from previous research. We hope that combining new features with the features in our thesis will have an improvement for the prediction.

More research should be done for phosphorylated Tyrosine to achieve a better result. In both the P.ELM and PPA data sets, the classification performance using the Tyrosine data set achieved the lowest results. Improvement of features extraction and selection for the Tyrosine data set is suggested to increase performance.

Bibliography

[1] L. A. Kelley, S. Mezulis, C. M. Yates, M. N. .. Wass and M. J. E. Sternberg, "The Phyre2 web portal for protein modeling, prediction and analysis," *Nature Protocols,* vol. 10, p. Nature Protocols, 2015.

[2] C. T. Walsh, S. Garneau-Tsodikova and G. J. J. Gatto, "Protein Posttranslational Modifications: The Chemistry," *Angewandte Chemie International Edition,* p. 7342–7372, 2005.

[3] N. Blom, T. Sicheritz-Pontén, R. Gupta, S. Gammeltoft and S. Brunak, "Prediction of post-translational glycosylation and phosphorylation of proteins from the amino acid sequence," *Proteomics,* vol. 4, no. 6, p. 1633–1649, 2004.

[4] R. Aebersold and M. Mann, "Mass spectrometry-based proteomics," *Nature,* vol. 422, pp. 198-207, 2003.

[5] H. Cao, L. J. Deterding, J. D. Venable, E. A. Kennington, J. R. Yates III, K. B. Tomer and P. J. Blackshear, "Identification of the anti-inflammatory protein tristetraprolin as a hyperphosphorylated protein by mass spectrometry and site-directed mutagenesis," *Biochemical Journal,* vol. 394, pp. 285-297, 2006.

[6] Y. Xue, Z. Liu, J. Cao, Q. Ma, X. Gao, Q. Wang, C. Jin, Y. Zhou, L. Wen and J. Ren, "GPS 2.1: enhanced prediction of kinase-specific phosphorylation sites with an algorithm of motif length selection," *Protein Engineering Design & Selection,* vol. 24, p. 255–260, 2011.

[7] Y. Dou, Y. Yao and Y. Zhang, "PhosphoSVM: Prediction of phosphorylation sites by integrating various protein sequence attributes with a support vector machine," *Amino Acids,* vol. 46, no. 6, p. 1459–1469, 2014.

[8] H. D. Ismail, A. Jones, J. H. Kim, J. H. Newman and D. B. .KC, "RF-Phos: A Novel General Phosphorylation Site Prediction Tool Based on Random Forest," *BioMed Research International,* vol. 2016, p. 12, 2016.

[9] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research,* vol. 3, pp. 1157-1182, 2003.

[10] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence,* vol. 97, pp. 273-324, 1997.

[11] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, no. 1, p. 5–32, 2001.

[12] V. N. Vapnik, Statistical Learning Theory, New York: A Wiley-Interscience Publication, 1998.

[13] H. Dinkel, C. Chica, C. Via, C. M. Gould, L. J. Jensen, T. J. Gibson and F. Diella, "Phospho.ELM: a database of phosphorylation sites—update 2011," *Nucleic Acids Research,* vol. 39, p. D261–D267, 2011.

[14] K. Sikic and O. Carugo, "Protein sequence redundancy reduction: comparison of various methods," *Bioinformation,* vol. 5, p. 234–239, 2010.

[15] P. Durek, R. Schmidt, J. L. Heazlewood, A. Jones, D. MacLean, A. Nagel, B. Kersten and W. X. Schulze, "PhosPhAt: the Arabidopsis thaliana phosphorylation site database. An update," *Nucleic Acids Research,* pp. D828-D834, 2010.

[16] H. B. Rao, F. Zhu, G. B. Yang, .. R. Li and Z. Chen, "Update of PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence," *Nucleic Acids Research,* vol. 39, p. W385–W390, 2011.

[17] M. Bhagwat and L. Aravind, "Chapter 10 PSI-BLAST Tutorial," in *Comparative Genomics,* vol. 1 and 2, N. Bergman, Ed., Totowa, New Jersey: Humana Press, 2007.

[18] N. Xiao, D.-S. Cao, M.-F. Zhu and Q.-S. Xu, "protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences," *Bioinformatics,* vol. 31, no. 11, pp. 1857-1859, 2015.

[19] M. Bhasin and G. P. S. Raghava, "Classification of Nuclear Receptors Based on Amino Acid Composition and Dipeptide Composition," *Journal of Biological Chemistry,* vol. 279, p. 23262–23266, 2004.

[20] Z. R. Li, H. H. Lin, L. Y. Han, L. Jiang, X. Chen and Y. Z. Chen, "PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence," *Nucleic Acids Research,* vol. 34, pp. W32-W37, 2006.

[21] I. Dubchak, I. Muchnik, C. Mayor, I. Dralyuk and S.-H. Kim, "Recognition of a protein fold in the context of the SCOP classification," *Proteins: Structure, Function, and Bioinformatics,* vol. 35, no. 4, p. 401–407, 1999.

[22] J. Shen, J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, Y. Li and H. Jiang, "Predicting protein–protein interactions based only on sequences information," *Proceedings of the National Academy of Sciences,* vol. 104, p. 4337–4341, 2007.

[23] B. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochimica et Biophysica Acta (BBA) - Protein Structure*, vol. 405, pp. 442-451, 1975.

[24] N. Blom, S. Gammeltoft and S. Brunak, "Sequence and structure-based prediction of eukaryotic protein phosphorylation sites," *Journal of Molecular Biology,* vol. 294, no. 5, p. 1351–1362, 1999.

[25] A. K. Biswas, N. Noman and A. R. Sikder, "Machine learning approach to predict protein phosphorylation sites by incorporating evolutionary information," *BMC Bioinformatcis,* vol. 11, no. 273, 2010.

[26] J. Gao, J. J. Thelen, A. K. Dunker and D. Xu, "Musite, a Tool for Global Prediction of General and Kinase-specific Phosphorylation Sites," *Molecular & Cellular Proteomics,* vol. 9, pp. 2586-2600, 2010.