EFFICACY OF DIFFERENT PROTEIN DESCRIPTORS IN PREDICTING PROTEIN FUNCTIONAL FAMILIES USING SUPPORT VECTOR MACHINE

ONG AI KIANG, SERENE

(B.Sc (Hons), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF PHARMACY

NATIONAL UNIVERSITY OF SINGAPORE

2007

ACKNOWLEDGMENTS

I would like to express my sincerest appreciation to my supervisor, Associate Professor Chen Yu Zong, for his excellent mentorship and counsel; I have learned a lot from his insightful advice.

I wish to also like to thank Dr Lin Hong Huang for his invaluable guidance and Dr Li Ze Rong, whose molecular descriptor program formed the basis for my own scripts.

I am also grateful to all members of the BIDD group, especially Zhiqun, Hailei, Xie Bin, Shuhui and (soon-to-be) Dr Cui Juan, who were not only lab-mates but dear friends as well.

Finally, this thesis is dedicated to my husband and partner.

TABLE OF CONTENTS

ABSTRACT

Sequence-derived structural and physicochemical descriptors have frequently been used in machine learning prediction of protein functional families; there is thus a need to comparatively evaluate the effectiveness of these descriptor-sets by using the same method and parameter optimization algorithm, and to examine whether the combined use of these descriptor-sets help to improve predictive performance. Six individual descriptor-sets and four combination-sets were evaluated in support vector machines (SVM) prediction of six protein functional families. While there is no overwhelmingly favourable choice of descriptor-sets, certain trends were found. The combination-sets tend to give slightly but consistently higher MCC values and thus overall best performance; in particular, three out of four combination-sets show slightly better performance compared to one out of six individual descriptor-sets. This study suggests that currently used descriptor-sets are generally useful for classifying proteins and that prediction performance may be enhanced by exploring combinations of descriptors.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DT	Decision tree
EC	Enzyme commission
FN	False negative
FP	False positive
GPCR	G-protein coupled receptors
*k*NN	*k* nearest neighbor
MCC	Matthews correlation coefficient
NN	Neural networks
OSH	Optimal separating hyperplane
QP	Quadratic programming
SLT	Statistical learning theory
SVM	Support vector machine
TN	True negative
TP	True positive

LIST OF PUBLICATIONS

A. Publications relating to research work from the current thesis

1. **Ong, A.K.S.**, H. H. Lin, Y.Z. Chen, Z.R. Li and Z.W. Cao, *Efficacy of different protein descriptors in predicting protein functional families*. BMC Bioinformatics, accepted, 2007.

B. Publications from other projects not included in the current thesis

1. Xie, B., C.J. Zheng, L. Y. Han, **S. Ong**, J. Cui, H.L. Zhang, L. Jiang, X. Chen and Y. Z. Chen, *PharmGED: Pharmacogenetic Effect Database*. Clin Pharmacol Ther, 2007. **81**(1): p. 29

2. Zheng C.J., L.Y.Han, B.Xie, C.Y.Liew, **S. Ong**, J.Cui, H.L.Zhang, Z.Q.Tang, S.H.Gan, L.Jiang and Y.Z. Chen, *PharmGED: Pharmacogenetic Effect Database*. Nuclei Acid Res, 2007. **35**(SI): p. D794–D799

1 INTRODUCTION

One of the more challenging and unsolved problems in current proteomics is that of protein functional prediction, and increasingly, various machine learning approaches are utilized towards solving this problem. The first section (Sec. 1.1) gives an overview of the biological problem and considers the various computational approaches, with a focus on machine learning methods. The second section (Sec. 1.2) introduces various machine learning approaches, and the last section (Sec. 1.3) gives the motivation and objective for this thesis.

1.1 Application of Machine Learning in Protein Functional Family Prediction

1.1.1 Biological importance of protein functional prediction

Proteins are involved in all of the processes that regulate the functional cycles of living organisms, performing a plethora of critical processes such as catalysis of biochemical reactions, transport of nutrients, recognition and transmission of signals. Thus, knowledge of protein function and interaction with other biomolecules is essential in a more fundamental understanding of biological phenomena such as gene regulation, disease pathology [1, 2], and cellular processes [3–6]. Though the genomes of over a hundred organisms are now known, the number of experimentally characterized proteins

lags far behind as traditional experimental techniques in determining protein structure and function such as X-ray diffraction or nuclear magnetic resonance methods, which remain difficult, costly and laborious; certainly they do not scale up to current sequencing speeds [7–10]. In addition, protein interactions and their native environments are highly complex and specific, which can make it difficult to replicate in the laboratory. As the sequencing of a growing number of genomes is completed, the gap between the flood of sequence information and their functional characterization is increasing rapidly [11, 12]. In current databases and sequencing projects, about 30% of proteins do not resemble any known sequence and have no assigned structure or function; another 20% were found to be homologous to a known sequence whose structure or function, or both, is largely unknown [10]. Computational biology is central in bridging this gap and the prediction of both protein structure and function are core unsolved problems in this area [13–18].

The prediction of protein function is the focus of many current studies; querying MEDLINE [19] with 'predict protein function' retrieves over 1000 papers from one year, of which the overwhelming majority describes single-case studies in which tools are combined in efforts to predict aspects of function for a particular protein or protein family [20].[1] The authors found that the most successful approaches tend to combine artificial intelligence tools such as neural networks (NN) and support vector machines (SVM) with evolutionary information derived from multiple alignments and aspects of protein structure. Commonly used computational methods can be broadly divided into sequence-based approaches, structure-based approaches and statistical learning approaches — most

[1] The paper by Rost *et al.* was dated 2003. A latest MEDLINE query accessed 12 June 2007 retrieved 2279 papers in 2006 with the same query terms, 2132 papers in 2005, and 1841 papers in 2004.

successful approaches are based on machine learning approaches such as SVM, which have been applied in a large number of applications such as computational gene finding [21], prediction of DNA active sites, sequence clustering and analysis of gene expression data [22].

1.1.2 The case for computational approaches

As mentioned earlier, with the vast amount of biological information being generated, it is inefficient, or even impossible, to rely only on human analysis; even the highly experimentally annotated *Caenorhabditis elegans* ORFeome was significantly enriched by computational gene predictions [23]. Moreover, there are problems that cannot be tackled with traditional experimental approaches; hence, we would have to turn to computational approaches.

Sequence-based approaches

Studies have shown a distinct relationship between functional similarity and sequence similarity [24] — this fact constitutes the basis of sequence-based approaches. For example, Pawlowski *et al.* [25] examined the EC enzyme classification and found a good correlation between sequence and functional similarity, and Ahmad *et al*. [26] found sequence composition to be sufficient in predicting binding site predictions with good accuracies.

Sequence-based methods include such as homology searching, clustering and pattern identification; the most common is sequence alignment. These methods hinge on

the tenet that proteins that are similar in sequence are more likely to be similar in structure and function, thus, they attempt to identify pairs of homologous proteins that share, because of common ancestry, similar structure and/or function.

In sequence alignment methods, sequences of the unknown function protein are aligned with sequences of known function proteins at various levels of identities; from the level of sequence similarity, the potential function of the unknown function protein can then be estimated. The Needleman–Wunsch algorithm was proposed in 1970 [9] to solve global pairwise sequence alignment, and the Smith–Waterman algorithm was introduced in 1981 [27] to find related regions within sequences. The emphasis of pairwise sequence alignment methods is on finding the best-matching piecewise local or global alignments of sequences, however, these dynamic programming algorithms are inefficient when applied to a large sequence database. Lipman and Pearson proposed the FASTA algorithm in 1985 [28], and this was later superseded by the BLAST algorithm in 1990 [29], which has since grown in popularity to become one of the most widely used bioinformatics program; the Institute for Scientific Information's *Web of Science* has reported that the original paper by Altschul *et al.* [30] was the most third highly cited paper published in the past two decades [31] and the most highly cited in the 1990s [32], underscoring the rising importance of bioinformatics research. Unlike dynamic programming algorithms, the FASTA and BLAST algorithms do not aim to optimize alignments between sequences but instead rely on heuristic strategies to find approximate solutions — the BLAST algorithm, which gave a good balance between computational speed and sensitivity, approximates the Smith–Waterman algorithm, and though it is

slightly less accurate than Smith–Waterman, it is over 50 times faster. PSI-BLAST ("Position Specific Iterated" BLAST), introduced in 1998, is an improvement upon the original BLAST and iteratively search protein databases for multiple alignments in order to find distant relatives and identify weak but biologically relevant similarities [33].

It is commonly observed in the literature that some regions within protein sequences are crucial for function and are thus better conserved among homologs as compared to surrounding regions [34, 35]. This led to the development of motif libraries such as Motifs [36] and Prosite [37], which catalog patterns repeatedly recurring in protein sequences.

However, there are drawbacks to a sequence-based approach. Not all homologous proteins have analogous functions [38]. Proteins with high sequence identity can fold into two different structures, hence giving different functionalities [39], and proteins with more than 30% sequence identity can adopt the same fold structures [40, 41]. In the absence of sequence similarities, particularly for proteins that are distantly related, this homology criterion becomes increasingly difficult to formulate [42]. It is also important to be aware of certain limitations and caveats when applying sequence alignment methods. Correlations thresholds between sequence similarity and functional similarity are a fundamental concern to groups utilizing sequence-based methods. In one study, Wilson *et al.* found that for pairs of domain that contain the same fold, precise function is usually conserved for sequence identity over 40%, approximately, and functional class is conserved for identity over 25% [43]. Generally, pairwise sequence identity is considered

high for alignments greater than 40%, and Doolittle has coined the term ‘twilight zone’ to describe the region with 20–30% identity as methods often fail to correctly align protein pairs in this range.

To complicate matters, the functional annotation of genomes remains an issue of contention [44–46] — Devos and Valencia found that up to 30% of the annotations might be erroneous [47] and Brenner reported that 8% of the annotations of the *Mycoplasma genitalium* genome in three published papers were in serious disagreement [48]. Thus, it is important to be aware of possible erroneous functional annotations that could have been introduced by the standard function prediction practice during the initial analysis.

Structure-based approaches

If sequence-based approaches can be thought of as utilizing one-dimensional information, then analogously, structure-based approaches rely on the analysis of two- and three-dimensional protein structures, under the assumption that proteins with similar structure have similar functions. Studies have found that proteins with similar sequences do adopt similar structures [49–52]; in fact, most protein pairs with more than 30% identity were found to be structurally similar [41]. Most sequence-based methods are based on the premise that there is an evolutionary relationship between sequences, thus, because structure is more conserved than sequence, structural information should enhance protein function prediction [53]. Families with low sequence identities (<30%) and yet have similar structural and functional characteristics are considered to possibly posses a common evolutionary origin, and such families are grouped into a superfamily [54]. Rost

et al. [20] have found that most successful approaches tend to contain evolutionary information derived from multiple alignments and aspects of protein structure.

In contrast to the effectiveness of sequence-based methods, structure alignment methods have uncovered homologous protein pairs with less than 10% pairwise sequence identity [55–57], and Rost [58] concluded that most similar protein structure pairs appear to have less than 12% pairwise sequence identity. Levitt and Gerstein [59] have found that structural comparison of protein pairs is able to detect approximately twice as many distant relationships as sequence comparison at the same error rate.

From shared protein folds, the function of an unknown protein could be deduced from existing structure-function knowledge of known proteins [60], and homology modelling approaches have been successfully implemented in this manner, by scanning new structures against a profile library [61–64]. The main limitation of this method is the restriction of sequence variation in the templates in the profile library. There are other drawbacks as well: (i) Knowledge of protein structures is necessary, and the gap between the number of sequences known and solved structures is increasingly rapidly to the extent that it becomes a serious limitation to the application of structure-based methods for predicting protein function — till now, the protein folding problem remains largely unsolved. Experimental methods to determine protein structures are time-consuming and have their own limitations, which in turn limits structure-based approaches [54, 65, 66]. *Ab initio* fold prediction methods can be applied to fill this gap, but they are computationally expensive and not as accurate [67]. (ii) Structure-based methods on their

own, without considering sequence similarity, are not very reliable [68–71]. (iii) Moreover, even if a group of proteins share a domain, it does not necessarily imply that these proteins have the same functionality [72, 73], for there are proteins with similar folds but no apparent sequence similarity, such collicins and globins [74].

Machine learning-based approaches

One restriction of sequence- and structure-based methods is that they require a certain level of similarity to exist (in sequence or structure). Also known as statistical learning approaches, the machine learning-based approaches are alternative methods that are not limited by this restriction, and while machine learning methods range from simple calculation of averages to the construction of complex models such as Bayesian networks, it is the latter end of the spectrum we are interested in for the purpose of this work, which includes methods such as naïve Bayes, C4.5 decision trees (DT), neural networks (NN) and support vector machines (SVM) [75]. Machine learning approaches aim to extract information from data through a process of training from examples. A certain number of representative examples, formed of positive samples from that specific functional class and negative samples of proteins outside of that functional class, are required to train a predictive model. Details of the theory as well as common methods will be elaborated in the subsequent section (Sec. 1.2).

There are advantages to a machine learning-based approach over the sequence- and structure-based approaches. For one, knowledge of the protein structure is not required, thus, these methods could be applied to cases in which the protein structure is

unknown or uncertain (highly flexible). Secondly, if the training samples are properly chosen and diverse, the predicted proteins will be more diverse as well. Thirdly, sequence similarity is not a requirement as some of these approaches are capable of utilizing only sequence-derived information.

However, there are still limitations to a purely statistical approach, for example, the *ab initio* prediction accuracy of tertiary structure from sequence alone remains unsatisfactory [76, 77], though interestingly, the best methods for protein *secondary* structure prediction are based on NN and SVM [78]. Furthermore, statistical approaches require accurate and sufficient training data, thus these methods are not applicable to problem domains that do not have enough pre-classified examples.

1.2 Introduction to Machine Learning

To take current definitions, machine learning is an area of artificial intelligence concerned with the development of techniques that allow computers to optimize a performance criterion using example data or past experiences [79]. The goal of machine learning is to extract useful information from data by building good probabilistic models, mimicking the human reasoning process [80]. Numerous algorithms have been developed and applied to a surprisingly wide variety of tasks, from engineering and science to business and commerce. There are several reasons why machine learning is important, for example, the ability to learn is a hallmark of intelligent behavior, so any attempt to understand intelligence as a phenomenon might help us to understand how animals and humans learn [81]. However, more pertinent to biological problems, there are other important reasons as well: (i) Some tasks cannot be defined well except by examples, for instance, input/output pairs might be specified exactly but not a concise relationship between input and output. Machine learning algorithms might be able to, given a large training dataset, produce a suitably constrained input/output function that approximates the implicit relationship. (ii) There could be important relationships and correlations masked within large volumes of data. Data mining algorithms attempt to extract these relationships. (iii) Often, the specifics of the intended working environment might not be completely known at the time of design, and machine learning methods can be used to refine performance. In this manner, machines can also be exported to different environments and optimized as well. Also, environments might change over time, and constant redesign is inefficient. (iv) The amount of data might be too large for explicit

coding by humans, for instance, as more and more genomes are sequenced. (v) And finally, learning provides a potential methodology for building high-performance systems [81, 82]. The application of machine learning is particularly important in areas where there is a large amount of data but little theory [22], such as bioinformatics.

The problem of protein family recognition studied in this work is essentially a problem of machine learning pattern recognition, though pattern recognition methods have found applications in diverse areas from data-mining, document classification and biometrics to financial forecasting. In particular, pattern recognition methods have recently gained increasing importance in bioinformatics in problems such as gene identification and protein differentiation. To define, pattern recognition is the study of how machines can observe the environment, learn to differentiate patterns of interest from their background, and make logical decisions about the categories of these patterns [83]. However, what constitutes a pattern? With reference to bioinformatics, a pattern may be a *motif* or a *fingerprint*, a particular sequence of amino acids or a specific set of physicochemical properties. In this study, amino acid sequences are represented as *descriptors* of various properties, and their recognition and classification are carried out by a machine learning algorithm.

1.2.1 Components of machine learning

A machine learning system essentially involves three main components, the choices of which are dictated by the problem domain: (i) data acquisition and pre-processing; (ii) data representation; and (iii) decision making or hypothesis. The problem should be well-

defined and sufficiently constrained (small intra-class variations and large inter-class variations), the data representation should be concise and the decision-making strategy simple [83]. Common issues regarding data and classifier are outlined below, while data representation — the main focus of this work — is introduced in greater detail in Sec. 1.3.

Most of the issues to consider in machine learning revolve around the data and choice of classifier. The data set should be sufficiently large and, as far as possible, balanced [84]. Many learning algorithms assume that the goal is to maximize accuracy and that the classifier will operate on data drawn from the same distribution as the training data; however, with these assumptions, if the data is unbalanced, unsatisfactory classifiers will be produced as training will be skewed towards the majority class. Fortunately, there are methods to deal with imbalanced data [85, 86]. Another issue is that of optimal complexity. Many methods suffer from underfitting or overfitting the data: underfitting occurs when the algorithm used does not have the capacity to express the variability in the data, while in overfitting, the algorithm has ‘too much capacity’ and therefore also ‘fits’ in noise present in the data. The cause for under- and overfitting depends on the complexity with which the model allows to express the variability in the data — if too much complexity is allowed, the variability due to noise is worked in as well; however, if the complexity is too low, the model will not be able to adequately represent the diversity of the data. Overfitting or underfitting also depends on the size of the training set — with small training sets, large deviations are possible and thus overfitting might occur [87].

As for the classifier, the machine learning algorithm should also have good predictive accuracy and robustness. It should also be reasonable fast and not require too much computational space. Linear classifiers are generally more robust than their non-linear counterparts as they have less free parameters to tune and are thus less prone to overfitting. Linear classifiers are also less affected by outliers or noise as compared to non-linear methods. The influence of outliers or noise can be tempered with methods such as regularization [88, 89]. Though a 'simple', i.e. linear, function that explains most of the data is generally preferable to non-linear functions that explain all of the data (Occam's razor), many practical problems are intrinsically non-linear in nature. In such a situation, a linear classifier in the appropriate kernel feature space, for example SVM, works well. Another desirable feature in machine learning algorithms is that of good generalization properties (good generalization refers to the model's ability to predict unseen data based on known learning data).

1.2.3 Categories of machine learning

Machine learning can be categorized based on the dataset. If the data used for learning is labeled, the problem becomes one of *supervised learning*, where the true label is known for a given set of data. Examples of such methods include *k*NN and SVM. If the labels are not known, then the problem is one of *unsupervised learning*, in which the aim is to characterize the structure of the data, for example by identifying groups of examples within the dataset that are collectively similar to each other (small intra-class distance) and distinct form other data (large inter-class distance). In other words, in supervised

learning, the classes are defined by the users or the system designer; but in unsupervised learning, the classes are learned based on the similarity of patterns. In supervised learning, the training data include training input and desired output and the task of the machine is to predict the value after being trained by the input samples. In contrast, there is no *a priori* output in unsupervised learning. All of the training examples are considered a set of random variables and treated evenly, and the model does not have any advance 'preconception' of the correct or incorrect answers. Furthermore, if the labels are categorical, the problem becomes that of *classification*; if the labels are continuously-valued, the problem is that of *regression* [10, 75, 83].

1.2.3 Overview and comparison of common machine learning algorithms

Decision trees

The decision tree (DT) [90–92] is one of the most popular machine learning algorithms and is often used in data mining and pattern recognition applications. It is used to identify the strategy most likely to reach a defined goal — which is to predict a category given an event — and compared to many of the other methods introduced in the succeeding subsections, it is simple to construct and efficient. A DT classifier separates the labeled points of the training data using hyperplanes that are perpendicular to one axis and parallel to all other axes, via a greedy algorithm that iteratively selects a partition whose entropy is greater than a given threshold, and then splits the partition to minimize this entropy by adding a hyperplane through it [93].

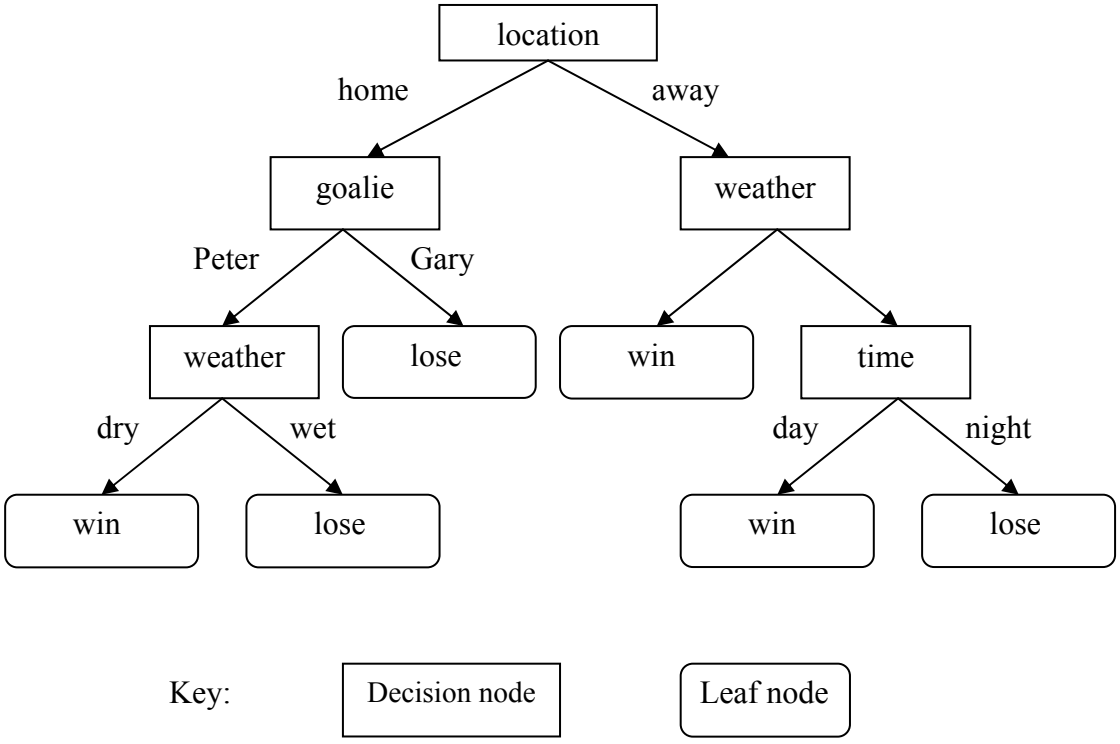

Figure 1: Example of a simple decision tree classification.

Given an instance of an object or situation, which is specified by a set of properties or attributes, the DT will return a 'yes' or 'no' decision about that instance. In other words, a DT is equivalent to a set of 'if-then' rules. DTs generate a series of rules from the training input samples, which are applied to the classification of unknown samples. These rules are linked in a tree structure, starting from the topmost node or root. Each node branches out into multiple nodes, and every decision at a node determines the direction of the next node movement, i.e. each leaf node is a Boolean classifier for that input instance. In this way, an optimal path is traced through the tree recursively until the bottommost node is reached. The DT is built top-down using *recursive partitioning* and it

should be: (i) consistent with the training data; and (ii) simple, in accordance with Occam's Razor. In simpler DTs, each node is usually based on more data and hence is more reliable. However, fully consistent DTs tend to over-generalize, especially if the DT is big or if there is not much training data, thus, there is a trade-off between full consistency and compactness — larger DTs can be more consistent but smaller DTs generalize better. The main aim of the DT algorithm is to select the attribute that contains the most information at each decision, i.e. the greatest improvement to the prediction accuracy [79, 92]. To achieve this, statistical properties such as *information gain* or *information gain ratio* are defined so as to quantitatively measure how well a given attribute separates the training samples according to their target classification.

The main advantages of DT are its speed and, in particular, its perspicuity (the ease with which the algorithm and its representation can be understood). With a DT, it is possible to interpret the decision rule in terms of individual features, i.e. the rules are human interpretable and can provide insights into the problem domains. This algorithm is good for tasks such as classification or predicting outcomes, or when the goal is assignment of a query to a few broad categories. Disadvantages with DTs include problems with sparse data and overfitting, moreover, DTs are not able to directly combine information about different features. Construction of the tree continues until all of the training examples are classified, however, noise or erroneous data are often present, leading to overfitting [79]. Overfitting may be alleviated by modifying the stopping criteria or by pruning the tree. There are numerous algorithms, depending on the level of interpretability desired, though C4.5 is the most popular choice [92].

k-nearest neighbors

Nearest-neighbor (NN) models date back at least to 1951 and have been a standard method in statistics and pattern recognition, where it is one of the oldest and simplest methods for performing general, non-parametric classification [94, 95]. The basis of the NN classifier is to choose the class of the nearest example in the training set, as measured by a distance metric, when classifying an unknown query. The *k*-nearest neighbors (*k*NN) is an extension of this idea, where the most common class of *k* nearest neighbors is chosen to classify the unknown query instead. *k*NN falls under what is known as instance-based learning programs, which learns by storing examples as points in a feature space and requires some means of measuring distance between examples [96]; *k*NN is also known as lazy learning, where the function is only approximated locally and all computation is deferred until classification. Instance-based methods have been applied to problems such as prediction of cancer recurrence, diagnosis of heart disease, prediction of protein secondary structure and prediction of DNA promoter sequences, and have been shown to compare favorable to other algorithms such as DTs on a wide range of domains in which feature values were either numeric or binary [96–98].

The *k*NN algorithm is fairly straightforward: the training examples are represented as vectors in a multidimensional feature space, which is partitioned into regions by the locations and class labels of the training samples. A query instance is then assigned to the class that is the most frequent class label among the *k* nearest neighbors. Thus, there are two components to the *k*NN algorithm: the similarity measure

(determination of neighbors), and the query assignment. The similarity between two points can be determined in several ways; the Euclidean distance is the most often used. Another important parameter to consider is the value of *k*. The best choice of *k* depends on the data; generally, larger *k* values reduce the effect of noise on the classification but make the boundaries between classes less distinct as the larger classes will overwhelm the smaller ones. However, one advantage of the *k*NN algorithm is that it is able to learn from a large training set, thus if *k* is set too small, the training model may not benefit from the large training set. The value of *k* should be greater than one (when k=1, this becomes the nearest neighbor algorithm) but less than N, where N is the size of the entire dataset; Dasarathy found that the ideal value of *k* is usually less than $\sqrt{N}$ [99], though the value of *k* in practice usually has to be estimated by cross-validation and then optimized through trial and error.

The main advantages of the *k*NN algorithm are its conceptual simplicity and its computational efficiency relative to other methods such as neural networks [79], and is a good choice when simplicity and accuracy are the predominant issues. Though simple, the algorithm returns good results and is used in numerous applications [100–103]. *k*NN is particularly suitable in cases where the training data set is very large and the dimension of the training vector is small (less than 20) [104], or when the training set is continually changing. Often, *k*NN is often used to pre-process data before applying more complicated methods. The performance of the nearest neighbor algorithm is also robust; as the amount of data approaches infinity, the *k*NN algorithm is guaranteed to approach the Bayes error rate, which is the minimum achievable error rate given the distribution of the data, for

some value of *k*, where *k* increases as a function of the number of data points [99]. Moreover, *k*NN can also be applied to datasets of continuous variables. However, the prediction accuracy of *k*NN can be badly affected by noise or irrelevant features, or if the feature scales are not consistent with their importance as the similarity measure considers *all* attributes from *all* training examples, thus, the selection or scaling of features to improve classification performance is an important research question [99].

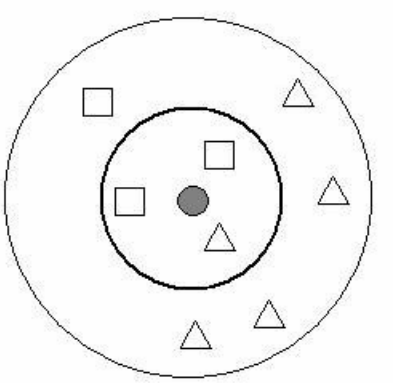

Figure 2: Example of a simple k Nearest Neighbour classification.

In Fig. 2, the query instance (shaded circle) will be classified as a square if *k* is small; on the other hand, it will be classified as a triangle if *k* is large.

Neural networks

Also sometimes known as *artificial neural networks,* a neural network (NN) is a linked group of simple processing element known as *neurons* that uses a mathematical model to process information based on a connectionist approach to computation. However, though these networks are called 'neural' in the sense that they were inspired by neuroscience and designed to emulate the central nervous system, they are not exact models of biologic

neural or cognitive phenomena, but instead tend to be more closely related to traditional mathematical and/or statistical models such as optimization algorithms and statistical regression models [105].

The concept of neural network dates back to 1957, when psychologist Frank Rosenblatt proposed a family of theoretical and experimental neural network models called perceptrons, which eventually set the foundations for important neural network models used today; this initial model of a learning machine marked the start of mathematical analysis of learning processes. Most crucially, the perceptron could learn, which was the breakthrough to pioneer today's current neural network technologies. A perceptron is, simply, a connected network that simulates an associative memory. Composed of an input layer and output layer of nodes, each of which are fully connected to the other with adjustable weights, this network will produce an output. The adjustment of the weights to produce a particular desired output is called 'training' the network and this is the mechanism that allows the network to learn [106, 107]. That early theory of perceptrons has its limitations but it sets the basis for future works, such as multi-layer networks. The basis of any modern neural network algorithm remains to incrementally adjust the network weights so as to improve a predefined performance measure over time, analogous to an optimization process. In other words, the learning process can be thought of as a 'search' in a multidimensional parameter (weight) space for a solution, which gradually optimizes a specified objective function or criterion. Learning can be supervised, in which the weights are gradually synthesized and updated until each input pattern or signal approaches its associated desired target pattern, or unsupervised, in

which some criterion or performance function is optimized until the weights and outputs of the network converge to representations that capture the statistical regularities of the input data. It is this adaptive or learning characteristic that makes these neural networks appealing in application domains where one has little or incomplete understanding of the problem to be solved but where training data is readily available. Neural networks have been applied to a wide array of problems including pattern classification, clustering [105], drug discovery [108], protein structure prediction [109] and protein function prediction [110].

Another key advantage of neural networks lies in their intrinsic parallelism, which allows for fast computation when these networks are implemented on parallel architectures [105]. Neural nets can also be extremely robust, if optimized well. They perform well on multivariate, non-linear domains, where other methods such as decision trees or rule induction system tend to falter [111].

There are significant disadvantages to using neural networks. Most neural net learning algorithms require significantly more time for training than other machine learning methods. Training is normally performed by repeatedly presenting the network with instances from a training set, and allowing it gradually to converge on the best set of weights for the task. For example, the training time for back-propagation, the most widely used neural net algorithm, is many orders of magnitude (10^2–10^4) greater than the training time for simpler algorithms such as ID3 [112–114]. Another important drawback of neural networks is that the model is implicit, hidden in the network structure and

optimized weights between the nodes. The individual relations between the input and output variables are not developed using an analytical basis so the model tends to be a black box [115, 116].

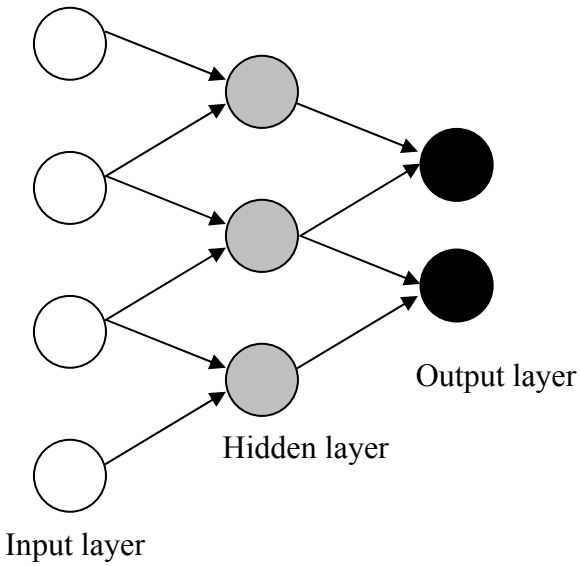

Figure 3: Example of a simple neural network.

Support vector machines

Support vector machines (SVM), which first proposed in the 1990s [117], fall under the umbrella of supervised machine learning methods of linear classifiers, which are used for binary classification (pattern recognition) and regression (real valued function approximation). SVM are called *discriminatory approaches* because they learn from the discrimination boundary; this is in contrast to learning a model for each class, for example, Bayesian classification, which is known as a *generative approach*. They have proven to be effective in diverse tasks from text categorization [118] and natural language processing [119] to bioinformatics, including cancer diagnosis [120], microarray gene

expression analysis [121], protein secondary structure prediction [122], protein-protein interaction prediction [123] and protein functional class prediction [7–9]. These studies have demonstrated that SVM is consistently superior to other supervised learning methods [7, 121, 124, 125].

Based on the structural risk minimization (SRM) principle from statistical learning theory (SLT) [126], SVM is a supervised learning classifier that maps data from feature vector space (input) to a higher dimensional class label space (output). Where classical statistics deal with large sample size problems, statistical learning theory is the first theory that is able to address also small sample learning problems. In this higher dimensional space, mathematical functions called kernel functions can be used to separate the two sample classes, and the goal of SVM is to select the optimal separating hyperplane. Details of SVM theory and mathematics can be found in Sec 2.1.

The primary advantage of SVM is that of good generalization; a theorem from SLT states that the choice of the maximum margin hyperplane (maximizing the margin of the training set) will minimize the generalization error, provided that the data is well-behaved [127]. Thus, SVM avoids the problem of overfitting in high dimensional space. They are also able to deal with data of very high dimensionality and are able to ignore irrelevant dimensions. Unlike DTs, SVM is able to rank properties. This ability to rank, as compared to classification alone, allows the generation of smaller output sets with higher relevance [93]. Another important advantageous property of the SVM algorithm is that it will definitely converge on a global solution because training an SVM amounts to

solving a convex quadratic programming problem, which also means that even if the solution is not unique, the set of global solutions is convex, and if the objective function is strictly convex, the solution is guaranteed to be unique [128, 129]. This means that SVM training will always find a global solution, as compared to neural networks, where many local minima usually exist. The SVM algorithm can also be extended to cope with noise in the training set and with multiple classes [130]. Linear classifiers are generally more robust, and a kernel-based method such as SVM means all of the advantages of linear classification (such as optimality) are maintained but the overall classification is non-linear in input space, since the feature and input space are non-linearly related, allowing classification to be performed on non-linear datasets [126].

The biggest limitation of the support vector approach lies in choice of the kernel. Once the kernel is decided, SVM classifiers have only one user-chosen parameter (the error penalty), but the kernel is a very big rug under which to sweep parameters. Work has been done on limiting kernels using prior knowledge [124, 131], but the best choice of kernel for a given problem is still a research issue. A second limitation is the (slow) speed and (huge) size of the SVM classifier, both in training and testing. While the speed problem in test phase is largely solved by Burges [124, 132], this still requires two training passes. Training for very large datasets (millions of support vectors) is an unsolved problem. Discrete data presents another problem, although excellent results have nevertheless been obtained with suitable rescaling [133]. Finally, SVM classifiers are intrinsically binary. They can be easily combined to handle a multiclass case, for

example, by training N one-versus-rest classifiers [134], however, an optimal design for multiclass SVM classifiers is still an open question [124].

1.3 Thesis Focus: Efficacy of Descriptors in Protein Functional Family Prediction

1.3.1 Role of descriptors

One of the most important components of machine learning algorithms is the descriptor, or feature. (The two terms are sometimes used interchangeably in the literature parlance.) These descriptors serve to represent and distinguish proteins or peptides of different structural, functional and interaction profiles by exploring their distinguished features in compositions, correlations, and distributions of the constituent amino acids and their structural and physicochemical properties [7, 135–137]. Sequence-derived structural and physicochemical descriptors have frequently been used in the machine learning prediction of protein structural and functional classes [7–9, 135, 138–140], protein-protein interactions [123, 137, 141], subcellular locations [142, 143] and peptide containing specific properties [138, 144].

In statistical pattern recognition, or classification, a pattern is represented by a set of d descriptors, giving a d-dimensional feature vector. Using statistical decision theory, decision boundaries between pattern classes are then established. As mentioned earlier, there are two parts to building a recognition model: training (learning) and testing (classification). In the training mode, feature extraction/selection is used to find the appropriate descriptors for representing the input patterns, and the classifier is trained to partition this feature space. In the testing mode, the trained classifier assigns the input

pattern to one of the pattern classes under consideration based on the measured features [83].

1.3.2 Types of descriptors

There are many ways to classify descriptors. Commonly, they may be classified by their dimensionality, such as 1-, 2- and 3D descriptors that encode information on chemical composition (1D), topology (2D), and shape and functionality (3D), respectively [145, 146]. Descriptors can also be classified as global, non-local and local, depending on the type of information they capture [147]; this is particularly useful when studying protein folding. They can also be classified by the type of information they encode, such as steric (molar refractivity), geometric (molecular surface area), electrostatic (charged polar surface area), and so on.

This study focuses on common individual and combinations of protein structural and physiochemical descriptors that can be derived from amino acid sequence: (i) composition descriptors; (ii) physicochemical and structural descriptors; (iii) autocorrelation descriptors, which describe the level of correlation between two objects (protein or peptide sequences) in terms of their specific structural or physicochemical property; and (iv) sequence order descriptors, which encode information about the amino acid distribution patterns of a specific physicochemical property along a protein or peptide sequence. Further details about each class of descriptors are given in Sec. 2.2.

1.3.3 Thesis motivation

As protein functional prediction in supervised learning methods hinges upon the discriminative features that map a protein's characteristics to structure or biological function, both the calculation and choice of descriptors are of fundamental importance. Many types of information may be extracted from sequence information; however, their relevance to the problem at hand is another matter. Inaccurate descriptors lower the prediction performance of the machine learning algorithm and non-informative features — even if they are accurate — add noise to the classification procedure, masking the information contained in the discriminating features [148]. There is thus a need to select descriptors.

There are also other reasons for the need to reduce the number of descriptors used. The choice of a classification algorithm depends on various factors, for example the amount of information available, however, no matter which classification algorithm is used, it must be trained using the available training samples. Thus, the performance of a classifier depends on both the number of training samples as well as the descriptors used to describe the samples. At the same time, the goal of building such a classification model is to recognize future test samples, which are likely to be different from the training samples; thus, optimizing a classifier to maximize its performance on the training set may not always result in a favorable prediction performance on a test set, a problem that is known as overtraining [127, 149]. Poor performance of a classification model in predicting test samples may be due to any of the following factors: (i) insufficient training samples; (ii) the number of descriptors is too large relative to the number of

training samples (curse of dimensionality) [150]; (iii) the number of unknown parameters associated with the classifier is too large; or (iv) overtraining [83].

Hence, the two main reasons for reducing the dimensionality of the input vector are computational cost and classification accuracy. The rationale for the former is evident; as for the latter, a smaller number of descriptors can alleviate the curse of dimensionality, particularly when the number of training samples is limited. However, a reduction in the number of descriptors may lead to a loss in discriminating ability and hence accuracy of the prediction model — the choice of descriptors is thus an important decision. Moreover, Watanabe has shown that it is possible to make two arbitrary patterns similar by encoding them with a sufficiently large number of redundant descriptors [151]. Therefore, the problem now lies in the selection of descriptors.

Descriptor selection or, as it is more commonly known, feature selection is a process commonly used in machine learning, whereby a subset that leads to the smallest classification error is chosen from a set of features or descriptors. Feature selection is sometimes necessary either because it is computationally infeasible to use all available features, or because of estimation problems that result when data samples are limited and yet the number of features is large (the so-called curse of dimensionality). With proteins, an additional complication may arise because of uncertainties in structural information. The objective of feature selection is three-fold: (i) improving the prediction performance of the predictors; (ii) providing faster and more cost-effective predictors; and (iii) providing a better understanding of the underlying process that generated the data.

As of 1997, most problems under study often did not use more than 40 features. Since then, however, domains involving hundreds to tens of thousands of features are now common. For example, in gene selection from microarray data, in which the usual classification task is to separate healthy patients from the sick; while the number of patients (examples) usually number fewer than 100, the number of variables in the raw data ranges from 6000 to 60,000 [152].

Note that there is a subtle difference between the terms *feature extraction* and *feature selection*, which are sometimes used interchangeably in the literature. Feature extraction algorithms create new features based on transformations or combinations of the original feature set, while feature selection algorithms aim to select the best subset of the input feature set, where the selected features retain their original physical interpretation. Feature extraction often precedes feature selection. Feature selection has two advantages over feature extraction: (i) it results in savings in computational cost since unnecessary features are discarded; and (ii) the selected features retain their original physical interpretation, which may be important for understanding the physical process that generates the discriminative patterns. On the other hand, even if the transformed features generated by feature extraction may not have a clear physical meaning, these transformed features may provide a better discriminative feature ability than the best set of selected features [83]. Though the focus of this work will lean more towards feature selection than extraction as the aim is to find descriptors that return the best performance, results from this study might be useful for future work on feature extraction as well.

Descriptors are usually chosen based on intuition and/or chemical knowledge; however, the large variety of descriptors available makes the task of descriptors selection a complex issue. Important questions to consider include: How should descriptors be selected for different applications such as machine learning methods, docking or similarity searching? Which types of descriptors perform 'best'; are there general preferences? Is there a universally preferred set of descriptors? Do complex descriptors perform better than simpler ones? What about combinations of descriptors? Selection of descriptors is often thus an arbitrary decision, for example based on favourable results reported by other researchers in the literature, or via a brute force method. However, the former may not be a reliable method as problem domains or classifiers may differ, and hence may not be directly applicable. Regarding the latter, feature selection methods can be extremely time-consuming, particularly for large datasets. For example, sequential backward selection (SBS), a common approach that removes one feature at a time until no improvement in the criterion function is obtained, scales on the order N^2 [153]. Moreover, there is little transparency in feature selection methods, which work like a 'black box'; descriptors are not selected on an analytical basis but on how well they score on a cost function.

There has been a lot of work has been done on the performance of chemical descriptors [154–162]. Generally, researchers have found that there is no preferred set of chemical descriptors; instead, performance is highly dependent on application and dataset. In many cases, 2D descriptors, and in particular structural keys, are sufficient in

capturing differences between compounds. Moreover, combinations of a few selected descriptors perform better than all descriptors together, and combinations of a limited number of structural keys and 2D descriptors perform better than any combination of descriptors and the entire set of keys. In contrast, research on protein descriptors is more fledging and these works tend to focus on optimizing combinations of descriptor sets and predictive methods, or on the selection of best descriptors for a specific dataset [163–165]. In summary, the importance of the role of descriptors in machine learning algorithms, the need to improve upon the selection of currently available descriptors and the lack of research especially pertaining to protein descriptors, all led to the motivation for this thesis.

1.3.4 Research objective and scope

As surmised from the preceding sections, there has been an increasing focus on the prediction of protein functional families, and particularly on using machine learning methods [20]. A review of the current literature showed that there have been no studies emphasizing a closer examination of various protein descriptor types in a manner independent of the problem domain, or one that objectively benchmarks various types of protein descriptors. Thus, there is a need to comparatively evaluate on a more fundamental level the effectiveness of commonly used descriptor sets for predicting different functional problems by using the same machine learning method and parameter optimization algorithm.

SVM was chosen as the machine learning method for protein function family prediction because it is a popular method that has consistently been shown better performances than other machine learning methods [121, 166]. As this work is intended as a benchmarking study of the performance of various sets of descriptors, other than automatic optimization of results that is an integral part of the SVM programs, such as sigma value scanning, no further attempt was made to optimize the prediction performance of any descriptor class or of any dataset by manually tuning the parameters. Hence, prediction results reported in this work might differ from those in reported studies. As it would be impossible to study an exhaustive array of protein functional families, six diverse families were chosen instead to form the problem domains for this work. It should be emphasized that the performance evaluation for the studied descriptor sets are based only on these six datasets and the conclusions from this study might not be readily extended to other datasets.

In Sec. 2, methods used in this work will be introduced. The descriptor sets performance are presented and discussed in Sec. 3, and concluding comments as well as suggestions for future work are given in Sec. 4.

2 METHODOLOGY

In this section, the theory underlying support vector machine, the mathematical calculations behind the descriptors-sets and the computational method in which the datasets are generated, are explained. The descriptors can be computed online at the PROFEAT website (http://jing.cz3.nus.edu.sg/cgi-bin/prof/prof.cgi), which is available freely for non-commercial use.

2.1 Support Vector Machines (SVM)

2.1.1 Linear case

In the simplest example, a two-dimensional grid of two types of data points that are separable linearly, SVM aims to draw a straight line so as to separate the two data classes (see Fig. 4). The training set is composed of n examples, represented as $\chi = \{(x_1, y_1),\ldots, (x_n, y_n)\}$, where the input $x_i \in R^N$ is a vector in feature space and the output $y_i \in \{-1, 1\}$ denotes its class label. Suppose there exists a hyperplane that separates the positive from the negative samples (a 'separating hyperplane'). The points x_i that lie on the hyperplane satisfy $w^T \cdot x_i + b = 0$, where w is a unit vector normal to the hyperplane and b is a parameter that minimizes the Euclidean norm $||w||^2$.

The outputs y_i is connected the inputs x_i by the functional dependence, or the decision function,

$$f(x) = sign(< w, x > + b), \quad (1)$$

where sign(u) = 1 if $u > 0$ and sign (u) = −1 if $u \leq 0$.

Geometrically, the data points are divided into two regions in the output space: a region where the output y_i takes the value 1 and a region where y_i takes the value −1; and these two regions are separated by the hyperplane H. As shown in Fig. 5, there are a number of possibilities in which a hyperplane can separate the two classes, thus the objective of SVM is to choose the optimal plane. Assuming that all new data points lie somewhere near the training data, the hyperplane should be chosen such that small shifts in data do not produce fluctuations in prediction results; therefore, the hyperplane that separates the two classes with the largest margin is expected to produce the best generalization performance. This hyperplane is known as the Optimal Separating Hyperplane (OSH) [126, 130].

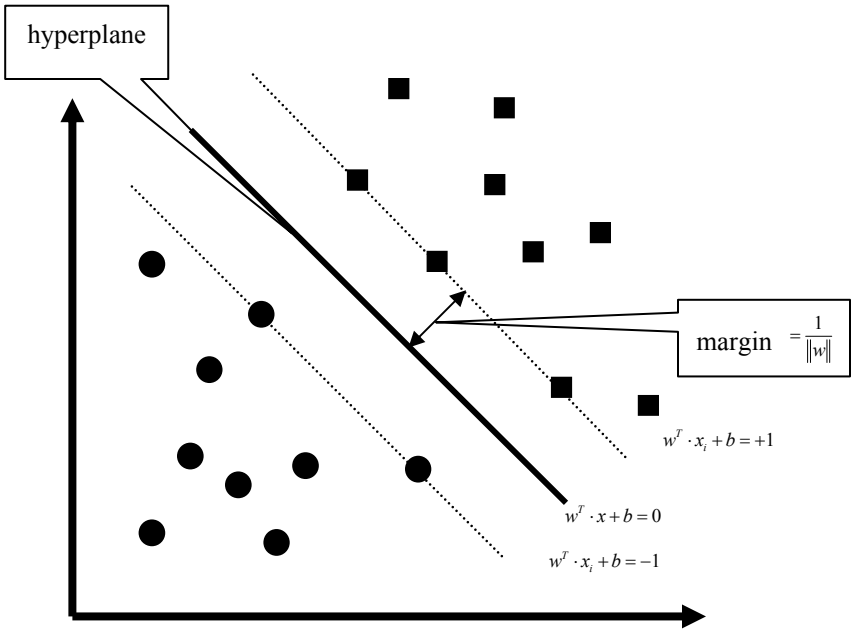

Figure 4: Finding a hyperplane to separate the positive and negative examples.

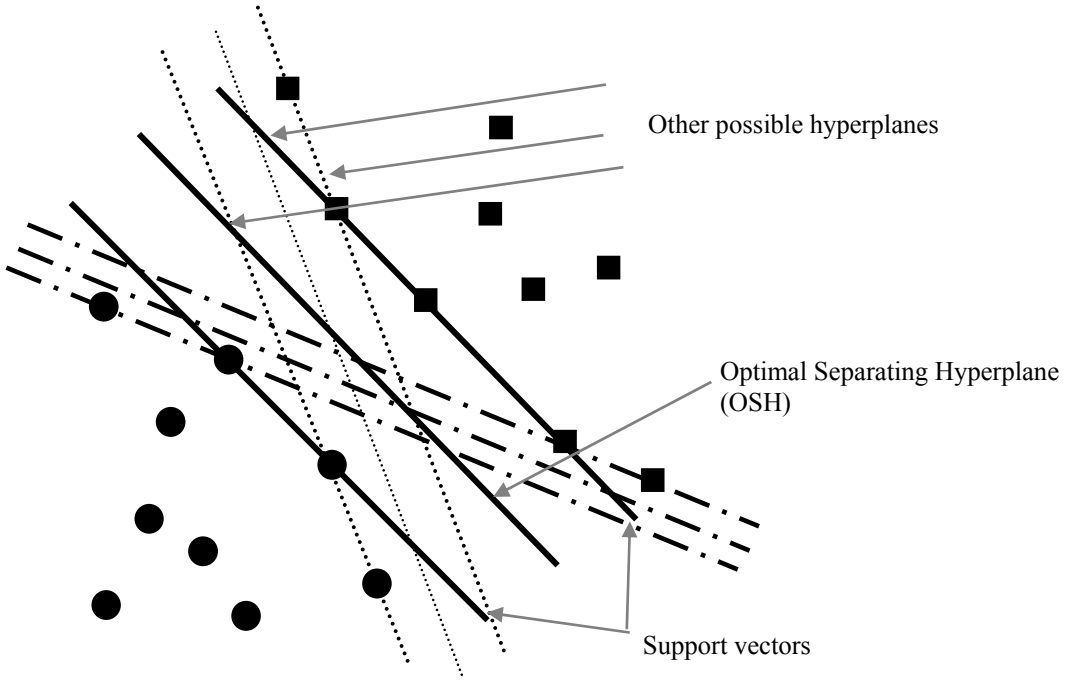

Figure 5: Optimal Separating Hyperplane (OSH).

Now, the margin $\gamma_i(w, b)$ of a training point x_i is defined as the distance between H and x_i

$$\gamma_i(w,b) = y_i(w \cdot x + b), \tag{2}$$

and the margin of a set of vectors $S = \{x_1, \ldots, x_n\}$ is defined as the minimum distance from H to the vectors in S

$$\gamma_X(w,b) = \min_{x_i \in S} \gamma_i(w,b) = \min_{\{x|y=1\}} \frac{w \cdot x}{\| w \|} - \max_{\{x|y=-1\}} \frac{w \cdot x}{\| w \|} \tag{3}$$

The OSH can be formulated as follows: suppose that all the training data satisfy the following constraints:

$$w \cdot x_i + b \geq 1 \text{ for } y_i = 1 \text{ (positive class)}, \tag{4}$$

$$w \cdot x_i + b \leq -1 \text{ for } y_i = -1 \text{ (negative class)}, \tag{5}$$

which can be combined into one set of inequalities

$$y_i(w \cdot x_i + b) \geq 1 \text{ for } i = 1, 2, \ldots, n. \tag{6}$$

Consider the points for which the equality (4) holds; these points lie on the hyperplane H_1: $x_i \cdot w + b = 1$, with normal w and perpendicular distance from the origin $|1-b|/\| w \|$. Similarly, the points for which Eq. (5) holds lie on the hyperplane H_2: $x_i \cdot w + b = -1$. Let d_+ (d_-) be the shortest distances from the separating hyperplane to the closest positive (negative) sample; $d_+ = d_- = 1/\|w\|$, and the margin is simply $2/\|w\|$. Note that H_1 and H_2 are parallel and that no training points fall between them. Thus, we can find the pair of hyperplanes that give the maximal margin (OSH) by minimizing $\|w\|^2$, subject the constraints (6). The training points that define these hyperplanes are known as *support*

vectors, as the removal of these points would change the solution. The OSH is, in fact, a linear combination of support vectors.

This optimization problem could be more efficiently solved by the Lagrange method. With the introduction of Lagrangian multipliers α_i (i = 1, … n), one for each of the inequality constraints, we obtain the Lagrangian

$$L_P = \frac{1}{2} \| w \|^2 - \sum_{i=1}^{l} \alpha_i y_i (x_i \cdot w + b) + \sum_{i=1}^{l} \alpha_i \,. \tag{7}$$

To solve the above, we would have to minimize L_P with respect to w and b, and simultaneously require that the derivatives of L_P with respect to the multipliers α_i vanish, subject to the constraints $\alpha_i \geq 0$. Requiring that the gradient of L_P with respect to w and b vanish leads to

$$w = \sum_i \alpha_i y_i x_i \,, \tag{8}$$

and

$$\sum_i \alpha_i y_y = 0 \,. \tag{9}$$

Substituting the above into Eq. (7), we get

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \,. \tag{10}$$

This particular dual formulation of the problem is called the Wolfe dual [129].[2] (Note that the Lagrangians are given different labels: P for primal, D for dual. The solution can

[2] This is a convex (QP) problem, since the objective function is itself convex, and the points that satisfy the constraints C_1 also form a convex set (any linear constraint defines a convex set, and a set of N simultaneous linear constraints defines the intersection of N convex sets, which is also a convex set). This

be found by minimizing L_P or by maximizing L_D.) The corresponding bias b_0 can be calculated as

$$b_0 = -\frac{1}{2}\left\{\min_{\{x|y=1\}}(w_0 \cdot x) - \max_{\{x|y=-1\}}(w_0 \cdot x)\right\}. \quad (11)$$

This quadratic programming (QP) problem can be solved efficiently through standard algorithms such as sequential minimization optimization (SMO) [167] or decomposition algorithms [168].

To sum, support vector training (for the linear separable case) amounts to maximizing L_D with respect to α_i, subject to constraints (9) and positivity of α_i, and the solution is given by (8). In the solution, the points for which $\alpha_i > 0$ are called the support vectors, and lie on one of the hyperplanes H_1, H_2. The support vectors are the most important elements in the training set; they lie closest to the decision boundary, and if all of the other training samples were removed, or moved around but not crossing H_1 or H_2, and the training was repeated, the same separating hyperplane would be found [124].

---

means that the solution can be obtained by equivalently solving the following 'dual' problem: *maximize* L_P, subject to the constraints that the gradients of L_P with respect to w and b vanish, and subject also to the constraints $\alpha_i \geq 0$.]

2.1.2 Non-linear case

In reality, however, most problems are non-linear. By introducing a kernel technique, which maps the input data to a higher dimensional feature space (see Fig. 6), yielding a non-linear decision boundary in input space, a linear classifier can be applied. A vector in n dimensions can be plotted and classified by a hyperplane of n–1 dimensions. The kernel trick is a method to convert a linear classifier algorithm into a non-linear algorithm by using a non-linear function known as a kernel to map the input vectors into a higher dimensional space. This makes a linear classification in the new feature space equivalent to the non-linear classification in the original input space.

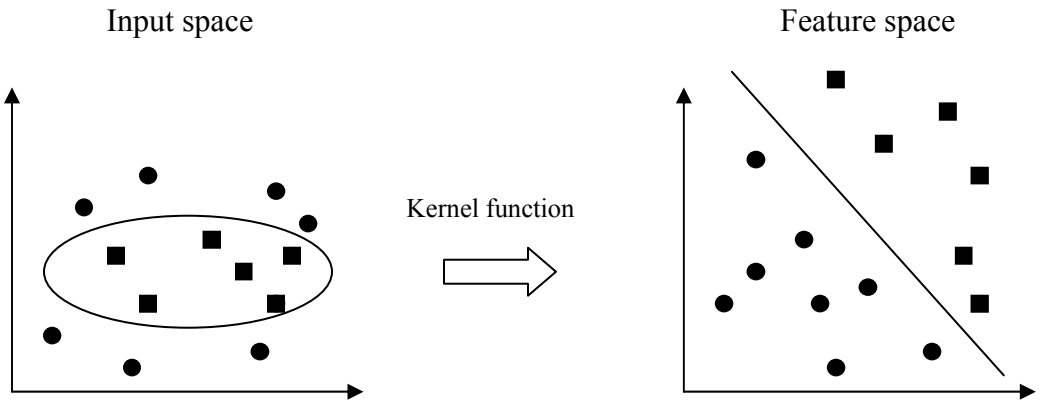

Figure 6: A kernel trick.

Let Φ denote an implicit mapping function from the input space to the feature space F. The, all of the above equations are transformed when we substitute x_i and the inner product in input space (x_i, x) by $\Phi(x_i)$ and the inner product kernel $K(x_i, x)$ respectively.

The kernel function is written as

$$K(x_i, x) = \Phi(x_i) \cdot \Phi(x) \tag{12}$$

and the Lagrangian (10) is now written as

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i \cdot x_j), \tag{13}$$

subject to the constraints $\sum_i \alpha_i y_y = 0$ and $\alpha_i \geq 0$ ($I = 1, 2, \ldots, n$). The bias b_0 is now

$$b_0 = -\frac{1}{2} \left\{ \min_{\{x|y=1\}} \left[\sum_{SV} \alpha_i y_i K(x_i, x) \right] - \max_{\{x|y=-1\}} \left[\sum_{SV} \alpha_i y_i K(x_i, x) \right] \right\} \tag{14}$$

and the decision function

$$f(x) = sign\left[\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b_0 \right] = sign\left[\sum_{SV}^{n} \alpha_i y_i K(x_i, x) + b_0 \right]. \tag{15}$$

In other words, the kernel technique transforms any algorithm that depends solely on the dot product between two vectors, replacing any dot product used with the kernel function. In this manner, a linear algorithm can be transformed into a non-linear algorithm. Note that the Φ function is never explicitly computed; this is important because it reduces the computational load and because the feature space may be infinitely dimensional, as is the case when the kernel is a Gaussian [169]. A function can be used as

a kernel function if and only if it satisfies Mercer's condition [170].[3] Well-known kernel functions include

Polynomial: $k(x,z) = (< x,z. + 1)^p$

Sigmoid: $k(x,z) = \tanh(\kappa < x,z > -\delta)$

Radial basis function (RBF): $k(x,z) = \exp\left(\frac{-\| x-z \|^2}{2\sigma^2}\right)$

In this work, the RBF kernel is used as it is the most popular kernel [171].

[3] Mercer's condition states that there exists a mapping Φ and an expansion $K(x_i,x) = \sum_i \Phi(x_i)\cdot\Phi(y_i)$ if and only if, for any $g(x)$ such that $\int g(x)^2 dx$ is finite, then $\int K(x,y)g(x)g(y)dxdy \geq 0$.

2.2 Calculation of Descriptor-sets

A total of ten descriptor-sets are examined in this work. The descriptors chosen as some commonly found in the literature, and can be computed from the PROFEAT server [136]

Six sets of individual descriptors and three combination-sets have been separately utilized in machine learning prediction of different protein functional and structural properties, all of which have shown impressive predictive performances [136, 172, 173]. The six individual sets are amino acid composition (Set D1) , dipeptide composition (Set D2) [27], Moreau–Broto autocorrelation (Set D3) [174, 175], Moran autocorrelation (Set D4) [176], Geary autocorrelation (Set D5) [177], and the composition, transition and distribution of structural physicochemical properties (Set D6) [139, 178]. The three combination-sets are quasi sequence order formed by weighted sums of amino acid compositions and physicochemical coupling correlations (Set D7) [142-144, 179], pseudo amino acid composition (PseAA) formed by weighted sums of amino acid compositions and physicochemical square correlations (Set D8) [172, 180], and the combination of amino acid and dipeptide compositions (Set D9) [27, 173]. Finally, we also consider a fourth combination-set that collects descriptor-sets D1 through D8 (Set D10). Details of the descriptor-sets are described below and summarized in Table 1.

Table 1: Protein descriptors commonly used for predicting protein functional families.

Sets	Descriptor-sets	No. of descriptors (properties)	No. of components	Type	Physicochemical properties	Refs
D1	Amino acid composition	1	20	Sequence composition		[27]
D2	Dipeptide composition	1	400	Sequence composition		[27]
D3	Normalized Moreau–Broto autocorrelation	8	240	Correlation of physicochemical properties	Hydrophobicity scale, average flexibility index, polarizability parameter, free energy of amino acid solution in water, residue accessible surface area, amino acid residue volume, steric parameters, relative mutability	[174, 175]
D4	Moran autocorrelation	8	240	Correlation of physicochemical properties	Hydrophobicity scale, average flexibility index, polarizability parameter, free energy of amino acid solution in water, residue accessible surface area, amino acid residue volume, steric parameters, relative mutability	[176]
D5	Geary autocorrelation	8	240	Square correlation of physicochemical properties	Hydrophobicity scale, average flexibility index, polarizability parameter, free energy of amino acid solution in water, residue accessible surface area, amino acid residue volume, steric parameters, relative mutability	[177]
D6	Descriptors of composition, transition and distribution	21	147	Distribution and variation of physicochemical properties	Hydrophobicity, Van der Waals volume, polarity, polarizability, charge, secondary structures, solvent accessibility	[7, 8, 123, 137{Cui, 2006 #64, 139, 140, 178, 181]}
D7	Quasi sequence order	4	160	Combination of sequence composition and correlation of physicochemical	Hydrophobicity, hydrophilicity, polarity, side-chain volume	[142, 143]
D8	Pseudo amino acid composition	3	298	Combination of sequence composition and square correlation of physicochemical	Hydrophobicity, hydrophilicity, side chain mass	[172, 180]
D9	Combination of amino acid and dipeptide composition	2	420	Combination of sequence compositions		[27]
D10	Combination of all eight sets of descriptors	54	1745	Combination of all sets		

2.2.1 Composition descriptors

Both amino acid (Set D1) and dipeptide composition (Set D2) are relatively simplistic protein sequence descriptors [182]. Amino acid composition has been used to predict secondary structural content [183]. They are also frequently used in combination for predicting protein fold and structural classes (accuracy 72–95%) [184, 185], functional classes (accuracy 83–97%) [27], and subcellular locations (accuracy 79–91%) [186, 187].

Set D1 Amino acid composition is defined as the fraction of each amino acid type in a sequence [27, 186]

$$f(r) = \frac{N_r}{N}, \tag{16}$$

where r = 1, 2, …, 20, N_r is the number of amino acid of type r, and N is the length of the sequence.

Set D2 Dipeptide composition is defined as

$$fr(r,s) = \frac{N_{rs}}{N-1}, \tag{17}$$

where r, s = 1, 2, …, 20, and N_{ij} is the number of dipeptides composed of amino acid types r and s. For 20×20 amino acid combinations, we obtain a vector containing 400 descriptor values.

2.2.2 Autocorrelation descriptors

Autocorrelation descriptors are a class of topological descriptors, also known as molecular connectivity indices, that describe the level of correlation between two objects (protein or peptide sequences) in terms of their specific structural or physicochemical property [174], which are defined based on the distribution of amino acid properties along the sequence [188]. Eight amino acid properties are used to derive the autocorrelation descriptors used in this work: (i) hydrophobicity scale, derived from the bulk hydrophobic character for the 20 types of amino acids in 60 protein structures [189]; (ii) average flexibility index derived from the statistical average of the B-factors of each type of amino acids in the available protein x-ray crystallographic structures [190]; (iii) polarizability parameter computed from the group molar refractivity values [191]; (iv) free energy of amino acid solution in water [191]; (v) residue accessible surface areas taken from average values of folded proteins [192]; (vi) amino acid residue volumes [193]; (vii) steric parameters derived form the van der Walls radii of amino acid side-chain atoms [194]; and (viii) relative mutability obtained by multiplying the number of observed mutations by the frequency of occurrence of the individual amino acids [195]. The amino acid indices were obtained from the Amino Acid index database (AAindex) [196]. Thus, each autocorrelation descriptor-set has 8 descriptors and 240 descriptor values, based on the parameter *d* set in the generation program.

In the literature, these descriptors have been used with good results. The Moreau–Broto autocorrelation descriptor [174, 175] (Set D3) has been applied in predicting transmembrane protein types (accuracy 82–94%) [197] and protein secondary structural

contents (accuracy 91–94%) [197]. The Moran autocorrelation descriptor [176] (Set D4) has been used in the prediction of protein helix contents (accuracy 85%) [198], and the Geary autocorrelation descriptor [177] (Set D5) has been utilized in analyzing allele frequencies and population structures [199].

Each of the properties is centralized and standardized such that

$$P_r' = (P_r - \overline{P}) / \sigma , \tag{18}$$

where $\overline{P}$ is the average of the property of the 20 amino acids. $\overline{P}$ and σ are given by

$$\overline{P} = \frac{\sum_{r=1}^{20} P_r}{20}, \tag{19}$$

and

$$\sigma = \sqrt{\frac{1}{20}\sum_{r=1}^{20}(P_r - \overline{P})^2} . \tag{20}$$

Set D3 Moreau–Broto autocorrelation descriptors are defined as [174, 175]

$$AC(d) = \sum_{i=1}^{N-d} P_i P_{i+d} , \tag{21}$$

where d = 1, 2, … , 30 is the lag of the autocorrelation, and P_i and P_{i+d} are the properties of the amino acid at positions i and $i+d$ respectively. After applying normalization, we get

$$ATS(d) = \frac{AC(d)}{N-d} . \tag{22}$$

Set D4 Moran autocorrelation descriptors are calculated as [176]

$$I(d) = \frac{\frac{1}{N-d}\sum_{i=1}^{N-d}(P_i - \overline{P})(P_{i+d} - \overline{P})}{\frac{1}{N}\sum_{i=1}^{N}(P_i - \overline{P})^2}, \quad (23)$$

where d, P_i and P_{i+d} are defined in the same way as that for Moreau–Broto autocorrelation and $\overline{P}$ is the average of the considered property P along the sequence:

$$\overline{P} = \frac{\sum_{i=1}^{N} P_i}{N}. \quad (24)$$

The Moran descriptor differs from that of the Moreau–Broto descriptor in that, instead of using property values, property deviations from the average values are utilized instead as the basis for measuring correlations.

Set D5 Geary autocorrelation descriptors are written as [177]

$$C(d) = \frac{\frac{1}{2(N-d)}\sum_{i=1}^{N-d}(P_i - P_{i+d})^2}{\frac{1}{N-1}\sum_{i=1}^{N}(P_i - \overline{P})^2}, \quad (25)$$

where d, $\overline{P}$, P_i and P_{i+d} are defined as above. This algorithm differs from the other two algorithms in its use of square-difference of property values instead of vector-product of property values or deviations as the basis for measuring correlations.

2.2.3 Composition, transition and distribution descriptors

In the Set D6, composition, transition and distribution descriptors represent the amino acid distribution patterns of a specific structural or physicochemical property along a protein or peptide sequence [139, 178], which have been used for the recognition of protein folds (accuracy 74–100%) [139] and the prediction of protein-protein interactions (accuracy 77–81%) [123, 137], protein functional families (accuracy 67–99%) [7, 8, 140, 181] and MHC-binding peptides (accuracy 97–99%) [138]. Seven types of physicochemical properties are considered in computing these features: (i) hydrophobicity; (ii) normalized van der Waals volume; (iii) polarity; (iv) polarizability; (v) charge; (vi) secondary structures; and (vii) solvent accessibility [7, 139, 178].

For each of these seven properties, the amino acids are divided into three groups based on the main amino acid indices clusters taken from Tomii and Kanehisa [139, 200] such that those in a particular group are regarded to have approximately the same property. The reason for dividing amino acids into three groups is that while amino acids can be divided into a minimum of both two and three groups for most attributes, they can only be divided into a minimum of three groups for attributes such as charge (positive, negative and neutral) and secondary structure (helix, strand and coil); therefore, the choice of three groups appears to be a more rational choice [7, 8, 123, 137, 139, 140]. The ranges of these numerical values and the division of the amino acids are shown in Table 2. The three descriptors: composition (*C*), transition (*T*) and distribution (*D*), are then computed for each attribute. The composition descriptor *C* is defined as the number of residues with that particular property divided by the total number of residues in a

protein sequence; it describes the global percent composition of each group of amino acids in a protein. *T* characterizes the percent frequency with which residues with a particular property is followed by residues of a different property, i.e. the percent frequencies with which the attribute changes its index along the entire length of the protein. *D* describes the distribution pattern of the attribute along the sequence by measuring the chain length within which the first, 25%, 50%, 75% and 100% of the amino acids with a particular property are located respectively.

For instance, consider the hydrophobicity attribute. Residues can be divided into hydrophobic (CVLIMFW), neutral (GASTPHY), and polar (RKEDQN) groups. The composition descriptor *C* consists of three values: the global percent compositions of (i) polar, (ii) neutral, and (iii) hydrophobic residues, in the protein. The transition descriptor *T* also consists of 3 values: the percent frequency with which (i) a polar residue is followed by a neutral residue or a neutral residue by a polar residue, (ii) a polar residue is followed by a hydrophobic residue or a hydrophobic residue by a polar residue, and (iii) a neutral residue is followed by a hydrophobic residue or a hydrophobic residue by a neutral residue. The distribution descriptor *D* consists of 5 values for each of the three groups: (i) the fractions of the entire sequence, (ii) the location of the first residue of a given group, and (iii) where 25%, 50%, 75%, and 100% of those are contained. Thus, there are 21 elements representing these three descriptors: 3 for *C*, 3 for *T* and 15 for *D*, and the protein feature vector is constructed by sequentially combining the 21 elements for all of these properties, resulting in a total of 7×21=147 dimensions. As an example,

consider a sequence MTEITAAMVKELRESTGAGA. According to the hydrophobicity division in Table 2, its hydrophobicity descriptor is encoded as 32132223311311222222.

Table 2: The division of amino acids into three groups for each attribute based on amino acid indices clusters.

Attribute	Divisions		
Hydrophobicity	Polar	Neutral	Hydrophobicity
	R,K,E,D,Q,N	G, A, S,T,P,H,Y	C,L,V,I,M,F,W
Normalized van der	Range 0–2.78	Range 2.95–4.0	Range 4.03–8.08
Waals volume	G,A,S,T,P,D	N,V,E,Q,I,L	M,H,K,F,R,Y,W
Polarity	Values 4.9–6.2	Values 8.0–9.2	Values 10.4–13.0
	L,I,F,W,C,M,V,Y	P,A,T,G,S	H,Q,R,K,N,E,D
Polarizability	Values 0–1.08	Values 0.128–0.186	Values 0.219–0.409
	G,A,S,D,T	C,P,N,V,E,Q,I,L	K,M,H,F,R,Y,W
Charge	Positive	Neutral	Negative
	KR	ANCQGHILMFPSTWYV	DE
Secondary structure	Helix	Strand	Coil
	EALMQKRH	VIYCWFT	GNPSD
Solvent accessibility	Buried	Exposed	Intermediate
	ALFCGIVW	PKQEND	MPSTHY

Composition descriptors Composition refers to the global percent for each encoded class in each sequence, and can be calculated as

$$C_r = \frac{n_r}{N} \ , \tag{26}$$

where r = 1, 2, 3, n_r is the number of r in the encoded sequence and N is the length of the sequence. In the same hydrophobicity division example (32132223311311222222), the number for encoded classes ‘1’, ‘2’ and ‘3’ are 5, 10 and 5, and the compositions are $\frac{5}{20} = 25\%$, $\frac{10}{20} = 50\%$ and $\frac{5}{20} = 25\%$, respectively.

Transition descriptors A transition from the index 1 to 2 is the percent frequency with which 1 is followed by 2 or 2 is followed by 1 in the encoded sequence, and is defined as

$$T_{rs} = \frac{n_{rs} + n_{sr}}{N - 1} , \tag{27}$$

where *rs* = '12', '13' and '23', n_{rs} and n_{sr} are the numbers of dipeptide encoded as '*rs*' and '*sr*' respectively in the sequence, and *N* is the length of the sequence.

Distribution descriptors This refers to the distribution of each attribute in the sequence. There are five distribution descriptors for each attribute and they are the position percents in the whole sequence for the first residue, 25% residues, 50% residues, 75% residues and 100% residues, respectively, for a specified encoded class. Consider the same hydrophobicity division example as above (32132223311311222222). There are 10 residues encoded as '2': the 2nd residue, 5th, and so on, and the distribution descriptors for '2' are 10.0 for the 2nd residue $\left(\frac{2}{20}\times 100\right)$, 25.0 for the 5th $\left(\frac{5}{20}\times 100\right)$, and so on.

2.2.4 Combination sets of amino acid composition and sequence order

Set D7 The quasi sequence order descriptors, proposed by Chou [142], are derived from both the Schineider–Wrede physicochemical distance matrix [143, 144, 186] and the Grantham chemical distance matrix [179] between each pair of the 20 amino acids. Four physicochemical properties are computed: (i) hydrophobicity, (ii) hydrophilicity, (iii) polarity, and (iv) side-chain volume. Similar to the descriptors in Set

D6, sequence order descriptors can also be used for representing amino acid distribution patterns of a specific physicochemical property along a protein or peptide sequence [144, 179]. For a protein chain of N amino acid residues $R_1R_2\ldots R_N$, the sequence order effect can be approximately reflected through a set of sequence order coupling numbers

$$\tau_d = \sum_{i=1}^{N-d} (d_{i,i+d})^2, \tag{28}$$

where τ_d is the dth rank sequence order coupling number (d = 1, 2,…, 30) that reflects the coupling mode between all of the most contiguous residues along a protein sequence, and $d_{i,i+d}$ is the distance between the two amino acids at position i and $i+d$. For each amino acid type, the first part of the quasi sequence order descriptor is defined as

$$X_r = \frac{f_r}{\sum_{r=1}^{20} f_r + w\sum_{d=1}^{30} \tau_d}, \tag{29}$$

where r = 1, 2, …, 20, f_r is the normalized occurrence of amino acid type i and w is a weighting factor (w = 0.1). The latter part of the quasi sequence order descriptor is defined as

$$X_d = \frac{w\tau_{d-20}}{\sum_{r=1}^{20} f_r + w\sum_{d=1}^{30} \tau_d}, \tag{30}$$

where d = 21, 22, …, 50. The combination of these two equations gives us a vector that describes a protein: the first 20 components reflect the effect of the amino acid composition, while the components from 21 to 50 reflect the effect of sequence order.

Set D8 The pseudo amino acid composition descriptor is actually an improvement upon the quasi sequence order descriptor [180]. Similar to the quasi-sequence order

descriptor, the pseudo amino acid descriptor (Set D8) is made up of a 50-dimensional vector in which the first 20 components reflect the effect of the amino acid composition and the remaining 30 components reflect the effect of sequence order, only now, the coupling number τ_d is now replaced by the sequence order correlation factor θ_λ. The set of sequence order correlated factors is defined as follows:

$$\theta_\lambda = \frac{1}{N-\lambda}\sum_{i=1}^{L-\lambda}\Theta(R_i, R_{i+\lambda}), \tag{31}$$

where θ_λ is the first-tier correlation factor that reflects the sequence order correlation between all of the λ-most contiguous resides along a protein chain (λ=1,…30) and N is the number of amino acid residues. $\Theta(R_i, R_j)$ is the correlation factor and is given by

$$\Theta(R_i, R_j) = \frac{1}{3}\left\{\left[H_1(R_j) - H_1(R_i)\right]^2 + \left[H_2(R_j) - H_2(R_i)\right]^2 + \left[M(R_j) - M(R_i)\right]^2\right\}, \tag{32}$$

where $H_1(R_i)$, $H_2(R_i)$ and $M(R_i)$ are the hydrophobicity [201], hydrophilicity [202], and side-chain mass of amino acid R_i, respectively. Before being substituted in the above equation, the various physicochemical properties $P(i)$ are subjected to a standard conversion,

$$P(i) = \frac{P^0(i) - \sum_{i=1}^{20}\frac{P^0(i)}{20}}{\sqrt{\frac{\sum_{i=1}^{20}\left[P^0(i) - \sum_{i=1}^{20}\frac{P^0(i)}{20}\right]^2}{20}}}. \tag{33}$$

This sequence order correlation definition [Eqs. (31) and (32)] introduces more physicochemical effects correlation factors as compared to the coupling number [Eq. (28)], and has shown to be an improvement on the way sequence order effect information

is represented [180, 203, 204]. Thus, for each amino acid type, the first part of the vector is defined as

$$X_r = \frac{f_r}{\sum_{r=1}^{20} f_r + w\sum_{d=1}^{30} \theta_j}, \tag{34}$$

where r = 1, 2, …, 20, f_r is the normalized occurrence of amino acid type i and w is a weighting factor (w = 0.1), and the second part is defined as

$$X_d = \frac{w\theta_{d-20}}{\sum_{r=1}^{20} f_r + w\sum_{d=1}^{30} \vartheta_\lambda}. \tag{35}$$

2.3 Protein Functional Families Datasets

The classification of proteins (by family) is widely accepted as an effective tool that can provide valuable insights into structure, activity and metabolic roles, and this organization of information is particularly important in the understanding of the vast amount of data from high-throughput genome projects. As a basic approach to large-scale genomic annotation, protein family classification has several advantages: (i) it improves the identification of proteins that are difficult to characterize based on pairwise alignments; (ii) it assists database maintenance by promoting family-based propagation of annotation and making annotation errors apparent; (iii) it provides an effective and efficient means to retrieve relevant biological information from vast amounts of data; and (iv) it reflects the underlying organization of gene families, the analysis of which is essential for comparative genomics and phylogenetics [205].

A number of different classification systems to organize proteins have been developed to address various annotation needs. To name a few of the most popular: (i) by protein domains, such as those in Pfam [206] and ProDom [207]; (ii) by hierarchical families, such as superfamilies/families [208] in the PIR-PSD, and protein groups in ProtoMap [209]; (iii) by sequence motifs or conserved regions, such as in PROSITE [210] and PRINTS [211]; (iv) by structural classes such as SCOP [212] and CATH [213]; or (v) through the integration of various family classifications such as ProClass/iProClass [214, 215] and InterPro [216]. In this work, the Pfam classification is used; i.e. proteins are classified by domains.

The protein functional families studied in this work include the enzyme EC 2.4 [203, 217–219], G-protein coupled receptors (GPCR) [220, 221], transporter TC 8.A [222, 223], chlorophyll [224], lipid synthesis proteins involved in lipid synthesis [225], and rRNA-binding proteins [140]. The dataset statistics are summarized in Table 3.

These six protein families were selected for testing the descriptor-sets based on their functional diversity, sample size and range of reported family member prediction accuracies [7, 135, 140, 181, 225]. Generally, these protein families play important roles in many cellular phenomenon vital to the proper functioning and regulation of living processes, and prediction of the functional roles are important not only in furthering our fundamental understanding of the mechanisms underlying various cellular processes, but also in the search for new therapeutic targets [226, 227]. The reported prediction accuracies for these families are generally lower than those of other families [8], which are ideal for critically evaluating the effectiveness of these descriptor-sets; having a lower accuracy should enable a better differentiation of the performance of the various classes. It should be noted that as SVM is essentially a statistical method, the datasets cannot be too small; yet it would also be convenient for the purposes of this study if they were not too large as to be unwieldy computationally.

2.3.1 Enzyme EC 2.4

The Enzyme Commission (EC) number is a numerical classification scheme for enzymes, based on the chemical reactions they catalyze. Note that since EC numbers do not specify enzymes but enzyme-catalyzed reactions, if different enzymes catalyze the same reaction, then they are assigned the same EC number. Each EC number has up to four components. For example, the class EC 2 refer to the tranferases, enzymes that facilitate the transfer of a functional group from one molecule to another; the sub-class EC 2.4 refer to the glycosyltransferases; EC 2.4.1 refer to hexosyltransferases; and EC 2.4.1.1 refers to phosphorylase or 1,4-α-D-glucan:phosphate α-D-glucosyltransferase, a specific hexosyltransferase.

The dataset studied in this work is the enzyme sub-class EC 2.4. Glycosyltransferases are enzymes that catalyze the synthesis of glycoconjugates through the transfer of a glycosyl moiety and are involved in post-translational modification of proteins (glycosylation). Metals such as magnesium or manganese are usually found in the active site and acts as a Lewis acid by binding to the di(phosphate) leaving group. Increased levels of glycosyltransferases have been found in disease states and inflammation [228, 229].

2.3.2 G-protein coupled receptors

G-protein coupled receptors (GPCR) are a large protein family of transmembrane receptors that transduct signals for inducing cellular responses. Ligands that bind and activate these receptors comprise a wide range, including light-sensitive compounds,

odors, pheromones, hormones and neurotransmitters, and vary in size from small molecules to large proteins. Members of GPCR are of great pharmacological importance, as 50–60% of approved drugs elicit their therapeutic effect by selectively addressing members of the GPCR family [230-233]. GPCR proteins are involved in just about every organ system and present a wide range of possible targets for diseases such as cancer, cardiac dysfunction, diabetes, central nervous system disorders, obesity, inflammation and pain [234].

2.3.3 Transporter TC 8.A

Transporters perform key roles in the transport of cellular molecules across cell and cellular compartment boundaries, mediating the absorption and removal of various molecules, including drugs, and regulating the concentration of metabolites and ionic species [235–237]. Functional transporter families are described according to the transporter classification (TC) system [222, 223] based on their mode of transport, energy coupling mechanism, molecular phylogeny and substrate specificity, particularly the first two characteristics as they are relatively stable [238]. Transporter families are classified based on five criteria, each corresponding to one of the five numbers or letters within the TC number, thus, a TC number (of a specific transporter protein) normally has five components (V.W.X.Y.Z): (i) V (a number 1, …, 9) corresponds to the transporter class; (ii) W (a letter A, B, …) corresponds to the transporter sub-class; (iii) X (a number) corresponds to the transporter family (sometimes actually a superfamily) under a sub-class; (iv) Y (a number) corresponds to the sub-family under a family, in which a transporter is found; and (v) Z represents the individual transporter under a sub-family.

The transporter sub-class TC 8.A studied in this work consists of auxiliary transport proteins, which are proteins that function or are complexed to known transport proteins, facilitating transport across membranes [222]. In particular, the TC 8.A sub-class comprise of proteins that in some way facilitate transport across one or more biological membranes but do not themselves participate directly in transport; these proteins always function in conjunction with one or more established transport systems. They may provide a function connected with energy coupling to transport, play a structural role in complex formation, serve a biogenic or stability function or function in regulation [222, 223].

2.3.4 Chlorophyll proteins

Chlorophyll proteins, a green photosynthetic pigment found in most plants, are essential for harvesting solar energy in photosynthetic antenna systems. Chlorophyll contains a porphyrin ring, a stable ring-shaped moiety around which electrons are free to migrate; thus, the ring has the ability to gain or lose electrons easily and hence provide energized electrons to other molecules — this is the fundamental process underlying photosynthesis. A magnesium ion is found in the center of the porphyrin ring, and the ring can have several different side chains [239].

2.3.5 Lipid synthesis proteins

Lipid synthesis proteins play central roles in processes such as metabolism and transport [240], cell signalling and membrane trafficking [241], and regulation of gene expression

and cell growth [242]. Deficiencies or altered functioning of lipid binding proteins are associated with disease states such as obesity, diabetes, atherosclerosis, hyperlipidemia and insulin resistance [240].

Lipid recognition by proteins is primarily mediated by some combination of a number of structural and physicochemical features including conserved fold elements [242], specific lipid-binding site architectures [243] and recognition motifs [244], ordered hydrophobic and polar contacts between lipid and protein [245], and multiple noncovalent interactions from protein residues to lipid head groups and hydrophobic tails [246].

2.3.6 rRNA binding proteins

Most cellular rRNAs work in concert with protein partners and protein-RNA interactions are critically important in the regulation of gene expression [6]; in particular, rRNA-binding proteins play central roles in the post-transcriptional regulation of gene expression [247, 248], and their binding capabilities are mediated by certain RNA binding domains and motifs [4, 249–251]. It is also known that binding of proteins to some catalytic RNA molecules will activate or enhance the activity of these molecules [252]. Correlated patterns of sequence and substructure, or motifs, in RNA-binding proteins have be shown to recognize and bind to specific RNA sequences and folds [253–255], patterns which a learning approach such as SVM are able to detect [123, 140]. Factors found to play roles in the recognition of RNA-binding proteins include amino acid composition and hydrophobicity (important considerations in the interaction of a

protein with other biomolecules) and charge and polarity (electrostatic interactions and hydrogen bonding to RNA, as the backbone is charged) [256].

2.4 Generation of Datasets

The datasets were obtained from SWISS-PROT [257], with the exception of TC 8.A, which was downloaded from the Transport Classification Database (TCDB) [258]; both are public databases. All distinct members in these downloaded datasets were used to construct the positive dataset for the corresponding SVM classification system; multiple entries were evenly distributed to the training, testing and independent evaluation sets.

Table 3: Summary of dataset statistics, including size of training, testing and independent evaluation sets, and average sequence length.

	Total		Training		Testing		Independent testing		Average sequence Size
	P	N	P	N	P	N	P	N	
EC 2.4	3304	14373	1382	5068	1022	5859	900	3446	460
GPCR	2819	21515	1580	7389	717	7333	522	6793	498
TC 8.A	229	23096	94	7962	72	7962	63	7172	483
Chlorophyll	999	22997	356	7928	333	7928	310	7141	480
Lipid	2192	11537	850	5779	707	4483	635	1275	312
rRNA	5855	13770	2004	5246	1940	4953	1911	3571	376

Next, the negative dataset, representing non-class members, is generated. It is impractical to include all proteins outside of a specific family as negative examples, thus, the approach to generative a comprehensive set of negative samples is to choose *representative* proteins from the all of the other protein families. Thus, each negative set (training, testing and independent) contains at least one randomly selected seed protein from each of the Pfam families, which number over 7000, in the PFAM database [206], and the representative proteins of these families *unrelated* to the protein family being

studied were chosen as negative samples. Thus, each training and testing negative set contains at least one randomly selected protein from each of the Pfam families. The size of the negative dataset is usually higher than that of the positive samples; this dataset imbalance explains why the negative prediction accuracy (specificity) is usually higher than the positive prediction accuracy (sensitivity).

These proteins, positive and negative, were further divided into separate training, testing and independent evaluation sets by the following procedure. First, proteins were converted into descriptor vectors and then clustered using hierarchical clustering into groups in the structural and physicochemical feature space [259], where more homologous sequences will have shorter distances between them, and the largest separation between clusters was set to a ceiling of 20. One representative protein was randomly selected from each group to form a training set that is sufficiently diverse and broadly distributed in the feature space. Another protein within the group was randomly selected to form the testing set. The selected proteins from each group were further checked to ensure that they are distinguished from the proteins in other groups. The remaining proteins were then designated as the independent evaluation set, also checked to be at a reasonable level of diversity. Fragments, defined as smaller than 60 residues, were discarded. This selection process ensures that the training, testing and evaluation sets constructed are sufficiently diverse and broadly distributed in the feature space. Though an analysis of the ‘similar’ proteins in each cluster showed that the majority of the proteins in a cluster are quite non-homologous, the program CDHIT (Cluster Database at High Identity with Tolerance) [260–262] was further used after the SVM

model was trained to remove redundancy at both 90% and 70% sequence identity, so as to avoid bias as far as possible. CDHIT removes homologous sequences by clustering the protein dataset at some user-defined sequence identity threshold, for example 90%, and then generating a database of only the cluster representatives, thus eliminating sequences with greater than 90% identity.

2.5 Performance Evaluation Methods

The aim of performance evaluation, of course, is to find out whether an algorithm has done well. Specifically in the case of prediction research, we want to know if a prediction algorithm is able to perform well on data that has not been used to construct the learning model, and the generalization capacity of the model to recognize new examples from the same data domain [263] — to do this, we would require comprehensive independent samples (the independent evaluation set).

As a discriminative method, the performance of SVM classification can be accessed by measuring the true positive *TP* (correctly predicted positive samples), false negative *FN* (positive samples incorrectly predicted as negative), true negative *TN* (correctly predicted negative samples), and false positive *FP* (negative samples incorrectly predicted as positive). As the numbers of positive and negative samples are imbalanced, the concepts of sensitivity and specificity are also introduced [124]. Sensitivity, or positive prediction accuracy, is the proportion of actual positives correctly predicted:

$$Q_P = \frac{TP}{(TP + FN)}. \tag{36}$$

Specificity, or negative prediction accuracy, is the proportion of actual negative correctly predicted:

$$Q_N = \frac{TN}{(TN + FP)}. \tag{37}$$

The overall accuracy is defined as

$$Q = \frac{TP + TN}{(TP + FN + TN + FP)}. \tag{38}$$

However, in some cases, Q, Q_P, and Q_N are insufficient to provide a complete assessment of the performance of a discriminative method, thus, the Matthews correlation coefficient (MCC) was chosen in this work to evaluate the randomness of the prediction [263, 264]:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}}, \tag{39}$$

where $MCC \in [-1,1]$, with a negative value indicating disagreement of the prediction and a positive value indicating agreement. A zero value means the prediction is completely random. The MCC utilizes all four basic elements of the accuracy and it provides a better summary of the prediction performance than the overall accuracy.

3 PERFORMANCE EVALUATION AND DISCUSSION

In this section, the performance of the various descriptor-sets are presented and discussed. Overall trends are first noted (Sec 3.1), and subsequent sections (Secs. 3.2–3.5) consider each of the various descriptor-sets, including the problem with using all available descriptors (Sec. 3.6).

3.1 Overall Trends

Independent validation datasets were used to test the prediction accuracies. Training and prediction statistics for the six datasets, across the ten studied descriptor-sets, are given in Table 4. The program CDHIT [129, 260–262] was used to remove redundancy at both 90% and 70% sequence identity so as to avoid bias; subsequently, the datasets are tested again with the independent evaluation sets and the statistics are given in Table 5.

Table 4: Dataset training statistics and prediction accuracies of six protein functional families. Predicted results given as TP (true positive), FN (false negative), TN (true negative), FP (false positive), Sen (sensitivity), Spec (specificity), Q (overall accuracy) and MCC (Matthews correlation coefficient).

Protein family	Descriptor set		Training set		Testing set				Independent evaluation set							
			P	N	P		N		P			N			Q(%)	MCC
					TP	FN	TN	FP	TP	FN	Sen(%)	TN	FP	Spec(%)		
EC 2.4	AA	D1	1249	2120	1154	1	9065	12	724	176	80.4	3244	202	94.1	91.3	0.74
	dipeptide	D2	1319	2120	1080	5	8806	1	646	154	82.9	3349	97	97.2	94.1	0.80
	Moreau-Broto	D3	1105	1756	1295	4	9166	5	768	132	85.3	3394	52	98.5	95.8	0.87
	Moran	D4	1239	2221	1161	4	8701	5	756	144	84.0	3365	81	97.7	94.8	0.84
	Geary	D5	1242	2223	1160	2	8690	14	753	147	83.6	3391	55	98.4	95.4	0.85
	C, T, D	D6	1214	2077	1145	45	8846	4	741	159	82.3	3383	63	98.2	94.9	0.84
	quasi	D7	1293	2624	1072	39	8295	8	696	204	77.3	3270	176	94.9	91.3	0.73
	pseAA	D8	1226	3008	1177	1	7918	1	794	106	88.2	3387	59	98.3	96.2	0.88
	1+2	D9	1275	2747	1129	0	8177	3	782	118	86.9	3367	79	97.7	95.5	0.86
	All	D10	1228	3254	1176	0	7672	1	798	102	88.7	3397	49	98.6	96.5	0.89
GPCR	AA	D1	1590	7458	1847	1	14166	3	505	17	96.7	6735	58	99.1	99.0	0.93
	dipeptide	D2	564	711	1728	3	14121	5	510	12	97.7	6737	56	99.2	99.1	0.93
	Moreau-Broto	D3	1169	4628	1122	4	10208	1	507	15	97.1	6737	56	99.2	99.0	0.93
	Moran	D4	1257	4474	1037	1	10363	0	499	23	95.6	6745	48	99.3	99.0	0.93
	Geary	D5	1290	4724	997	8	10113	0	494	28	94.6	6734	59	99.1	98.8	0.91
	C, T, D	D6	757	2060	1536	2	12777	0	503	19	96.3	6742	51	99.2	99.0	0.93
	quasi	D7	812	2950	1482	1	11887	0	495	27	94.8	6696	97	98.6	98.3	0.88
	pseAA	D8	653	2171	1644	0	12550	1	501	21	96.0	6769	24	99.7	99.4	0.95
	1+2	D9	1590	7458	693	12	7322	57	512	10	98.1	6735	58	99.1	99.1	0.93
	All	D10	672	2454	1625	0	12268	0	502	20	96.2	6757	36	99.5	99.2	0.94
TC 8.A	AA	D1	118	2858	49	0	13121	0	36	27	57.1	1843	2	99.9	98.5	0.73
	dipeptide	D2	116	1100	50	0	14824	0	41	22	65.1	1843	2	99.9	98.7	0.78
	Moreau-Broto	D3	94	7962	53	0	14501	0	42	21	66.7	1842	3	98.6	98.7	0.78
	Moran	D4	94	7962	47	0	11250	0	37	26	58.7	1843	2	99.9	98.5	0.74
	Geary	D5	94	7962	47	0	11137	0	37	26	58.7	1843	2	99.9	98.5	0.74
	C, T, D	D6	94	7962	64	0	15283	0	44	19	69.8	1843	2	99.9	98.9	0.81
	quasi	D7	94	7962	59	0	15045	0	43	20	68.3	1843	2	99.9	98.9	0.80
	pseAA	D8	103	943	63	0	14981	0	48	15	76.2	1843	2	99.9	99.1	0.85
	1+2	D9	114	810	52	0	15114	0	41	22	65.1	1843	2	99.9	98.7	0.78
	All	D10	102	1068	64	0	14856	0	48	15	76.2	1843	2	99.9	99.1	0.85

Table 4 (continued)

Protein family	Descriptor set		Training set		Testing set				Independent evaluation set						Q(%)	MCC
			P	N	P		N		P			N				
					TP	FN	TN	FP	TP	FN	Sen(%)	TN	FP	Spec(%)		
EC 2.4	AA	D1	1249	2120	1154	1	9065	12	724	176	80.4	3244	202	94.1	91.3	0.74
	dipeptide	D2	1319	2120	1080	5	8806	1	646	154	82.9	3349	97	97.2	94.1	0.80
	Moreau-Broto	D3	1105	1756	1295	4	9166	5	768	132	85.3	3394	52	98.5	95.8	0.87
	Moran	D4	1239	2221	1161	4	8701	5	756	144	84.0	3365	81	97.7	94.8	0.84
	Geary	D5	1242	2223	1160	2	8690	14	753	147	83.6	3391	55	98.4	95.4	0.85
	C, T, D	D6	1214	2077	1145	45	8846	4	741	159	82.3	3383	63	98.2	94.9	0.84
	quasi	D7	1293	2624	1072	39	8295	8	696	204	77.3	3270	176	94.9	91.3	0.73
	pseAA	D8	1226	3008	1177	1	7918	1	794	106	88.2	3387	59	98.3	96.2	0.88
	1+2	D9	1275	2747	1129	0	8177	3	782	118	86.9	3367	79	97.7	95.5	0.86
	All	D10	1228	3254	1176	0	7672	1	798	102	88.7	3397	49	98.6	96.5	0.89
GPCR	AA	D1	1590	7458	1847	1	14166	3	505	17	96.7	6735	58	99.1	99.0	0.93
	dipeptide	D2	564	711	1728	3	14121	5	510	12	97.7	6737	56	99.2	99.1	0.93
	Moreau-Broto	D3	1169	4628	1122	4	10208	1	507	15	97.1	6737	56	99.2	99.0	0.93
	Moran	D4	1257	4474	1037	1	10363	0	499	23	95.6	6745	48	99.3	99.0	0.93
	Geary	D5	1290	4724	997	8	10113	0	494	28	94.6	6734	59	99.1	98.8	0.91
	C, T, D	D6	757	2060	1536	2	12777	0	503	19	96.3	6742	51	99.2	99.0	0.93
	quasi	D7	812	2950	1482	1	11887	0	495	27	94.8	6696	97	98.6	98.3	0.88
	pseAA	D8	653	2171	1644	0	12550	1	501	21	96.0	6769	24	99.7	99.4	0.95
	1+2	D9	1590	7458	693	12	7322	57	512	10	98.1	6735	58	99.1	99.1	0.93
	All	D10	672	2454	1625	0	12268	0	502	20	96.2	6757	36	99.5	99.2	0.94
TC 8.A	AA	D1	118	2858	49	0	13121	0	36	27	57.1	1843	2	99.9	98.5	0.73
	dipeptide	D2	116	1100	50	0	14824	0	41	22	65.1	1843	2	99.9	98.7	0.78
	Moreau-Broto	D3	94	7962	53	0	14501	0	42	21	66.7	1842	3	98.6	98.7	0.78
	Moran	D4	94	7962	47	0	11250	0	37	26	58.7	1843	2	99.9	98.5	0.74
	Geary	D5	94	7962	47	0	11137	0	37	26	58.7	1843	2	99.9	98.5	0.74
	C, T, D	D6	94	7962	64	0	15283	0	44	19	69.8	1843	2	99.9	98.9	0.81
	quasi	D7	94	7962	59	0	15045	0	43	20	68.3	1843	2	99.9	98.9	0.80
	pseAA	D8	103	943	63	0	14981	0	48	15	76.2	1843	2	99.9	99.1	0.85
	1+2	D9	114	810	52	0	15114	0	41	22	65.1	1843	2	99.9	98.7	0.78
	All	D10	102	1068	64	0	14856	0	48	15	76.2	1843	2	99.9	99.1	0.85

Table 5: Dataset statistics and prediction accuracies after homologous sequences removal (HSR) at 90% and 70% identity. Predicted results given as TP (true positive), FN (false negative), TN (true negative), FP (false positive), Sen (sensitivity), Spec (specificity), Q (overall accuracy) and MCC (Matthews correlation coefficient).

Protein family	% HSR*	Descriptor set		Independent evaluation set							
				P			N			Q (%)	MCC
				TP	FN	Sen(%)	TN	FP	Spec(%)		
EC 2.4	90	AA	D1	552	250	68.8	3235	201	94.2	89.4	0.65
		dipeptide	D2	626	176	78.1	3339	97	97.2	93.6	0.78
		Moreau-Broto	D3	609	193	75.9	3384	52	98.5	94.2	0.80
		Moran	D4	603	199	75.2	3355	81	97.6	93.4	0.78
		Geary	D5	591	211	73.7	3381	55	98.4	93.7	0.79
		C, T, D	D6	501	301	62.5	3374	62	98.2	91.4	0.70
		quasi	D7	545	257	68.0	3261	175	94.9	89.8	0.66
		pseAA	D8	666	136	83.0	3375	61	98.2	95.4	0.84
		1+2	D9	630	172	78.6	3357	79	97.7	94.1	0.80
		All	D10	670	132	83.5	3388	48	98.6	95.8	0.86
	70	AA	D1	459	223	67.3	3193	199	94.1	89.6	0.62
		dipeptide	D2	516	166	75.7	3296	96	97.2	93.6	0.76
		Moreau-Broto	D3	503	179	73.8	3341	51	98.5	94.4	0.78
		Moran	D4	495	187	72.6	3311	81	97.6	93.4	0.75
		Geary	D5	484	198	71.0	3339	53	98.4	93.8	0.77
		C, T, D	D6	399	283	58.5	3330	62	98.2	91.5	0.67
		quasi	D7	452	230	66.3	3218	174	94.9	90.1	0.63
		pseAA	D8	551	131	80.8	3331	61	98.2	95.3	0.83
		1+2	D9	520	162	76.3	3314	78	97.7	94.1	0.78
		All (1–8)	D10	554	128	81.2	3344	48	98.6	95.7	0.84
GPCR	90	AA	D1	391	13	96.8	6724	58	99.1	99.0	0.91
		dipeptide	D2	395	9	97.8	6744	38	99.4	99.4	0.94
		Moreau-Broto	D3	393	11	97.3	6726	56	99.2	99.1	0.92
		Moran	D4	386	18	95.5	6734	48	99.3	99.1	0.92
		Geary	D5	381	23	94.3	6723	59	99.1	98.9	0.90
		C, T, D	D6	391	13	96.8	6731	51	99.3	99.1	0.92
		quasi	D7	382	22	94.6	6685	97	98.6	98.3	0.86
		pseAA	D8	387	17	95.8	6758	24	99.7	99.4	0.95
		1+2	D9	391	13	96.8	6752	30	99.6	99.4	0.94
		All (1–8)	D10	388	16	96.0	6762	20	99.7	99.5	0.95
	70	AA	D1	307	8	97.5	6695	58	99.1	99.1	0.90
		dipeptide	D2	309	6	98.1	6715	38	99.4	99.4	0.93
		Moreau-Broto	D3	306	9	97.1	6697	56	99.2	99.1	0.90
		Moran	D4	301	14	95.6	6705	48	99.3	99.1	0.90
		Geary	D5	198	17	94.6	6694	59	99.1	98.9	0.88
		C, T, D	D6	307	8	97.5	6702	51	99.2	99.2	0.91
		quasi	D7	296	19	94.0	6656	97	98.6	98.4	0.83
		pseAA	D8	301	14	95.6	6729	24	99.6	99.5	0.94
		1+2	D9	307	8	97.5	6723	30	99.6	99.5	0.94
		All (1–8)	D10	302	13	95.9	6733	20	99.7	99.5	0.95

Table 5 (continued)

Protein family	% HSR*	Descriptor set		Independent evaluation set							
				P			N			Q (%)	MCC
				TP	FN	Sen(%)	TN	FP	Spec(%)		
TC 8.A	90	AA	D1	28	27	50.9	1846	2	99.9	98.5	0.68
		dipeptide	D2	33	22	60.0	1846	2	99.9	98.7	0.75
		Moreau-Broto	D3	34	21	61.8	1845	3	99.8	98.7	0.75
		Moran	D4	29	26	52.7	1845	3	99.8	98.8	0.75
		Geary	D5	29	26	52.7	1845	3	99.8	98.8	0.75
		C, T, D	D6	36	19	65.5	1846	2	99.9	98.9	0.78
		quasi	D7	35	20	63.6	1845	3	99.8	98.8	0.76
		pseAA	D8	40	15	72.7	1845	3	99.8	99.2	0.82
		1+2	D9	33	22	60.0	1846	2	99.9	98.7	0.75
		All (1–8)	D10	40	15	72.7	1845	3	99.8	99.2	0.82
	70	AA	D1	25	24	51.0	1828	2	99.9	98.6	0.68
		dipeptide	D2	29	20	59.2	1828	2	99.9	98.8	0.74
		Moreau-Broto	D3	29	20	59.2	1827	3	99.8	98.8	0.73
		Moran	D4	26	23	53.1	1828	2	99.9	98.7	0.70
		Geary	D5	26	23	53.1	1828	2	99.9	98.7	0.70
		C, T, D	D6	33	16	67.3	1828	2	99.9	99.0	0.79
		quasi	D7	30	19	61.2	1827	3	99.8	98.8	0.74
		pseAA	D8	36	13	73.5	1827	3	99.8	99.2	0.82
		1+2	D9	29	20	59.2	1828	2	99.9	98.8	0.74
		All (1–8)	D10	36	13	73.5	1827	3	99.8	99.2	0.82
Chlorophyll	90	AA	D1	159	127	55.6	1594	8	99.5	92.9	0.70
		dipeptide	D2	205	81	71.7	1598	4	99.8	95.5	0.82
		Moreau-Broto	D3	224	62	78.3	1599	3	99.8	96.6	0.86
		Moran	D4	222	64	77.6	1599	3	99.8	96.5	0.86
		Geary	D5	211	75	73.8	1598	4	99.8	95.8	0.83
		C, T, D	D6	182	104	63.6	1594	8	99.5	94.1	0.75
		quasi	D7	159	127	55.6	1595	9	99.4	92.8	0.69
		pseAA	D8	233	53	81.5	1595	7	99.6	96.8	0.87
		1+2	D9	224	62	78.3	1594	8	99.5	96.3	0.85
		All (1–8)	D10	229	57	80.1	1597	5	99.7	96.7	0.87
	70	AA	D1	113	118	48.9	1578	8	99.5	93.1	0.65
		dipeptide	D2	155	76	67.1	1582	4	99.8	95.6	0.79
		Moreau-Broto	D3	171	60	74.0	1583	3	99.8	96.5	0.84
		Moran	D4	171	60	74.0	1583	3	99.8	96.5	0.84
		Geary	D5	161	70	69.7	1582	4	99.8	95.9	0.81
		C, T, D	D6	137	94	59.3	1578	8	99.5	94.4	0.72
		quasi	D7	114	117	49.4	1575	11	99.3	93.0	0.64
		pseAA	D8	182	49	78.8	1579	7	99.6	96.9	0.85
		1+2	D9	172	59	74.5	1578	8	99.5	96.3	0.82
		All (1–8)	D10	178	53	77.1	1581	5	99.7	96.8	0.85

Table 5 (continued)

Protein family	% HSR*	Descriptor set		Independent evaluation set							
				P			N			Q (%)	MCC
				TP	FN	Sen(%)	TN	FP	Spec(%)		
Lipid synthesis	90	AA	D1	403	149	73.0	1213	59	95.4	88.6	0.72
		dipeptide	D2	431	121	78.1	1256	16	98.7	92.5	0.81
		Moreau-Broto	D3	436	116	79.0	1268	4	99.7	93.4	0.84
		Moran	D4	421	131	76.3	1270	2	99.8	92.7	0.83
		Geary	D5	416	136	75.4	1270	2	99.8	92.4	0.82
		C, T, D	D6	449	103	81.3	1270	2	99.8	94.2	0.86
		quasi	D7	435	117	78.8	1269	3	99.8	93.4	0.84
		pseAA	D8	423	129	76.6	1265	7	99.5	92.5	0.82
		1+2	D9	449	103	81.3	1245	27	97.9	92.9	0.83
		All (1–8)	D10	454	98	82.3	1265	7	99.5	94.2	0.86
	70	AA	D1	316	138	69.6	1205	59	95.3	88.5	0.69
		dipeptide	D2	343	111	75.6	1248	16	98.7	92.6	0.81
		Moreau-Broto	D3	340	114	74.9	1260	4	99.7	93.1	0.82
		Moran	D4	330	124	72.7	1262	2	99.8	92.7	0.81
		Geary	D5	328	126	72.3	1260	4	99.7	92.4	0.80
		C, T, D	D6	358	96	78.9	1244	20	98.4	93.3	0.82
		quasi	D7	342	112	75.3	1257	7	99.5	93.1	0.82
		pseAA	D8	331	123	72.9	1257	7	99.4	92.4	0.80
		1+2	D9	360	94	79.3	1237	27	97.9	93.0	0.81
		All (1–8)	D10	360	94	79.3	1257	7	99.5	94.1	0.85
rRNA binding	90	AA	D1	1407	91	93.9	3502	59	98.3	97.0	0.93
		dipeptide	D2	1437	61	95.9	3510	51	98.6	97.8	0.95
		Moreau-Broto	D3	1403	95	93.7	3529	32	99.1	97.5	0.93
		Moran	D4	1347	151	89.9	3491	70	98.0	95.6	0.89
		Geary	D5	1347	151	89.9	3533	28	99.2	96.5	0.91
		C, T, D	D6	1451	47	96.9	3537	24	99.3	98.6	0.97
		quasi	D7	1358	140	90.7	3429	132	96.3	94.6	0.87
		pseAA	D8	1442	56	96.3	3531	30	99.2	98.3	0.96
		1+2	D9	1436	62	95.9	3518	43	98.8	97.9	0.95
		All (1–8)	D10	1449	49	96.7	3537	24	99.3	98.6	0.97
	70	AA	D1	924	83	91.8	3454	59	98.3	96.9	0.91
		dipeptide	D2	952	55	94.5	3463	50	98.6	97.7	0.93
		Moreau-Broto	D3	920	87	91.4	3483	30	99.2	97.4	0.92
		Moran	D4	907	100	90.1	3444	69	98.0	96.3	0.89
		Geary	D5	908	99	90.2	3485	28	99.2	97.2	0.92
		C, T, D	D6	963	44	95.6	3493	20	99.4	98.6	0.96
		quasi	D7	917	90	91.1	3382	131	96.3	95.1	0.86
		pseAA	D8	654	53	94.7	3484	29	99.2	98.2	0.95
		1+2	D9	950	57	94.3	3471	42	98.8	97.8	0.94
		All (1–8)	D10	960	47	95.3	3490	23	99.4	98.5	0.96

As the purpose of this work is to benchmark the performance of various descriptor-sets, it is helpful to perform an initial rough validation of the results from this work against other studies. For the dataset EC 2.4, the sensitivities, specificities and overall accuracies for the ten descriptor-sets ranged from 77.3–88.7%, 94.1–98.6% and 91.3–96.5%, respectively. Differences in the choice of descriptors notwithstanding, these results are comparable to previously reported SVM prediction results for EC 2.4: Cai *et al.* [8] reported sensitivity, specificity and overall accuracy values of 70.5%, 94.2% and 92.9% respectively. Similarly, a comparison of the SVM prediction results obtained from this work using a variety of different descriptor-sets with those previously reported in the literature [7, 8, 135, 140, 181, 225] is given in Table 6. Though the descriptors used in this study are not the same as those in the above-mentioned literature, by and large, the prediction results from this work agree with those in the literature. Note that the other studies were focused on developing a prediction system for a specific protein family such as enzymes, and did not consider various sets of descriptors, as in this work.

It is also observed that the prediction accuracies in this study for the non-members of a dataset (specificity) are always better than those for the members of a dataset (sensitivity). This is due to the way the negative training set is generated: as a highly diverse set of non-members for each dataset can be generated from the Pfam database, which comprise of over 8000 protein families [206], this results in a larger number of negative samples (Table 3), and hence, the SVM models were more comprehensively trained for the recognition of non-members. Moreover, this imbalance between the positive and negative training datasets tends to skew the SVM hyperplane closer to the

side with a smaller number of samples, which can lead to a lower prediction accuracy for those samples as compared to those on the other side of the hyperplane [265]. This is known as overfitting; it occurs when there is a large set of possible hypotheses and thus the learning algorithm can end up finding meaningless regularities. However, the size of the negative dataset cannot be simply reduced to match that of the positive dataset since this compromises the diversity required to fully represent all non-members in the feature space. There are computational methods to compensate for this imbalance [266], but as the focus of this current work is a comparative evaluation of different descriptor-sets, there was no need to employ such measures.

Table 6: Comparison of range of prediction accuracies for 10 descriptor-sets with others reported in the literature (highlighted in grey).

Dataset	Source	Sensitivity (%)	Specificity (%)	Overall accuracy (%)
EC 2.4		77.3–88.7	94.1–98.6	91.3–96.5
	90% HSR	62.5–83.5	94.2–98.6	89.4–95.8
	70% HSR	58.5–81.2	94.1–98.6	89.6–95.7
	Cai *et al.* [8]	70.5	94.2	92.9
GPCR		94.6–98.1	98.6–99.7	98.3–99.4
	90% HSR	94.6–97.8	98.6–99.7	98.3–99.5
	70% HSR	94.0–97.5	98.6–99.7	98.4–99.5
	Cai *et al.* [7]	95.6	98.1	97.4
TC 8.A		57.1–76.2	98.6–99.9	98.5–99.1
	90% HSR	50.9–72.7	99.8–99.9	98.5–99.2
	70% HSR	51.1–73.5	99.8–99.9	98.6–99.2
	Lin *et al.* [181]	74.3	99.8	99.5
Chlorophyll		57.4–82.3	99.3–99.9	92.7–96.9
	90% HSR	55.6–81.5	99.4–99.8	92.9–96.8
	70% HSR	48.9–78.8	99.5–99.8	93.0–96.9
	Cai *et al.* [7]	97.4	99.8	99.7
Lipid synthesis		74.0–83.6	95.5–99.8	88.4–94.2
	90% HSR	73.0–82.3	95.4–99.8	88.6–94.2
	70% HSR	69.6–79.3	95.3–99.5	88.5–94.1
	Lin *et al.* [225]	82.2	99.6	98.1
rRNA binding		93.3–97.9	97.4–99.3	96.0–98.6
	90% HSR	89.9–96.9	98.0–99.3	94.6–98.6
	70% HSR	90.1–95.3	98.3–99.4	95.1–98.6
	Han *et al.* [140]	94.1	98.7	98.6

As explained in Sec. 2.5, Matthews correlation coefficient (MCC) values were used instead of standard accuracy because it is a more robust measure of performance. The performances of the ten descriptor-sets were ranked by the MCC values of the respective SVM prediction of the six functional families, which are given in Table 7. The computed MCC scores for these descriptor-sets are in the range of 0.65–0.97 (90% homologous sequence removal, or HSR) and 0.62–0.96 (70% HSR) for all protein families studied. Accordingly, the performance of these descriptor-sets is categorized into two groups based on their MCC values: 'Exceptional' (>0.85) and 'Good' (≤0.85). At the same time, these descriptor-sets are aligned in the order of their MCC values with "=" being of equal values and ">" indicating that one is better than the other. It is noted that, as the differences of many of these MCC values are rather small, such alignment is likely superficial to some extent and may not best reflect the real ranking of performance. Overall, the performances of these descriptor-sets are not significantly different, there is no overwhelmingly preferred descriptor-set, and SVM prediction performance appears to be highly dependent on the dataset.

As shown in Tables 4 and 5, for many of the studied datasets, the differences in prediction accuracies and MCC values between different descriptor-sets are small. In particular, for GPCR and rRNA binding proteins, the results of almost all descriptor-sets are in the 'Exceptional' category. Examining the range of MCC values of the descriptor-sets for each of the studied protein families (70% HSR), the differences between the largest and smallest MCC values are, in order of increasing magnitude: 0.10, 0.12, 0.14, 0.16, 0.21 and 0.21 for rRNA binding proteins, GPCR, TC 8.A, lipid synthesis proteins,

Table 7: Descriptor sets ranked and grouped by MCC (Matthews correlation coefficient), before and after removal of homologous sequences at 90% and 70% identity, respectively.

Protein family	% HRS*	Prediction performance	
		Exceptional > 0.85	**Good** ≤ 0.85
EC 2.4	NR	D10 > D8> D9 > D3	D5 > D4=D6 > D2 > D1 > D7
	90%	D10	D8 > D3=D9 > D5 > D2=D4 > D6 > D7 > D1
	70%		D10 > D8 > D3=D9 > D5 > D2 > D4 > D6 > D7 > D1
GPCR	NR	D8 > D10 > D1=D2=D3=D4=D6=D9 > D5 > D7	
	90%	D8=D10 > D2=D9 > D3=D4=D6 > D1 > D5 > D7	
	70%	D10 > D8=D9 > D2 > D6 > D1=D3=D4 > D5	D7
TC 8.A	NR		D8=D10 > D6 > D7 > D2=D3=D9 > D4=D5 > D1
	90%		D8=D10 > D6 > D7 > D2=D3 =D4=D5=D9 > D1
	70%		D8=D10 > D6 > D2=D7=D9 > D3 > D4=D5 > D1
Chlorophyll	NR	D8=D10 > D4 > D3=D9	D5 > D2 > D6 > D7 > D1
	90%	D8=D10 > D3=D4	D9 > D5 > D2 > D6 > D1 > D7
	70%		D8=D10 > D3=D4 > D9 > D5 > D2 > D6 > D1 > D7
Lipid synthesis	NR	D10 > D6	D7 > D2=D3=D9 > D4=D8 > D5 > D1
	90%	D6=D10	D3=D7 > D4=D9 > D5=D8 > D2 > D1
	70%		D10 > D3=D6=D7 > D2=D4=D9 > D5=D8 > D1
rRNA binding	NR	D10 > D8=D9 > D2=D3=D6 > D1 > D7> D4=D5	
	90%	D6=D10 > D8 > D2=D9 > D1=D3 > D5 > D4> D7	
	70%	D6=D10> D8 > D9 > D2 > D3=D5 > D1 > D4 > D7	

chlorophyll proteins and EC.2.4 families respectively. Given that a difference of 0.10 and 0.20 in MCC values translates to an approximate 4% and 7% difference in overall prediction accuracy, this separation is not large indeed.

Though the dataset is a more important determinant of prediction performance than the choice of descriptor class, a few general trends could be observed. Three out of four of the combination-sets tend to exhibit slightly but consistently higher MCC values for the protein families studied in this work. These sets are Sets D8, D9 and D10. In contrast, only one out of six individual sets, Set D6, tend to exhibit slightly but consistently higher MCC values for the protein families studied in this work. Therefore, statistically speaking, it appears that the use of combination-sets tend to give slightly better prediction performance than the use of individual-sets.

3.2 Composition Descriptors

It was found that the combination of amino acid composition and dipeptide composition (Set D9) tend to give consistently better results than that of the individual descriptor-sets (Set D1 and Set D2). It is known that one drawback of amino acid composition descriptors is that the same amino acid composition may correspond to diverse sequences as sequence order is lost [27, 173], and this sequence order information can be partially covered by considering dipeptide composition (Set D2). On the other hand, dipeptide composition lacks information concerning the fraction of the individual residue in the sequence, thus, a combination-set is expected to give better prediction results, which has

been confirmed by this and other studies [27, 173]. The consistently poor performance of Set D1 alone suggests that composition information alone is not sufficient to completely distinguish different proteins, and though the use of sequence order information does improve prediction results significantly, the combination of these two types of information (Set D9) returned the best performance. In fact, Set D9 often returned a much better or comparable prediction performance than more complex descriptors such as autocorrelation.

3.3 Autocorrelation Descriptors

Autocorrelation descriptor sets were expected to perform well as they should be better able to capture distinguishing motifs unique to a protein functional family; however, as seen from the moderate performance by the autocorrelation sets, this was not the case. The only dataset in which they showed slightly better results was the chlorophyll proteins dataset; it is possible that this is due to the porphyrin ring in chlorophyll proteins, which consists of four nitrogen atoms binding strongly to a coordinated magnesium atom in a square planar arrangement [239], for this unique feature suggests that geometrical and/or topological descriptors should perform well.

3.4 Composition, Transition and Distribution Descriptors

Like the autocorrelation sets, the performance of the composition, transition and distribution (CTD) descriptors (Set D6) tend to fall in the middle when ranked, though

this set displayed more consistent results. CTD descriptors are often used in protein folding studies [267, 268] as they are good at representing the amino acid distribution patterns of a specific structural or physicochemical property along a protein or peptide sequence [139, 178], though they have also shown good results in protein functional prediction [7, 8, 140, 181]. It is also noted that Set D6 showed better performance in the lipid synthesis proteins and transporter TC 8.A datasets, both of which contain distinctive functional groups; lipid synthesis proteins are highly polar [246] while TC 8.A proteins are auxiliary transport proteins that have to bind to a diverse range substrates that include complex polysaccharides (TC 8.A.3) and sugars (TC 8.A.7) as well as metal ions (TC 8.A.11) [258].

3.5 Quasi Sequence Order and Pseudo Amino Acid Descriptors

Interestingly, though the combination sets D7 and D8 are similar, with the exception of lipid synthesis proteins, Set D8 always outperforms Set D7. In fact, while Set D8 is one of the top performers in this study, Set D7 ranks as one of the worst. The quasi sequence order descriptors (Set D7) were first proposed by Chou in 2000 [142] and takes in account both the amino acid composition as well as the effect of sequence order; subsequently, pseudo amino acid composition descriptors (Set D8) were introduced as an improvement upon the quasi sequence order descriptors. The basic method to extract composition and sequence order information from sequence remains the same, but the definition of pseudo amino acid composition can introduce more correlation factors of physicochemical effects [204]. Chou [142] found that the newer pseudo amino acid

descriptors improved prediction accuracy of nine membrane protein locations by about 15%, twelve subcellular locations by about 5% and five types of membrane proteins by about 5%. Though the type of information used is largely the same in both descriptor-sets, it is noteworthy that it is the difference in autocorrelation algorithms can lead to significantly different prediction results.

3.6 Entire Descriptor Set

Obviously, the inclusion of all descriptors is inefficient and computationally expensive, but more importantly, it was found that using all of the descriptor-sets (Set D10) generally, but not always, gives the best result. This is consistent with the findings on the use of molecular descriptors for predicting compounds of specific properties [160, 161], as well as studies on the use of feature selection methods [148, 269, 270]. For instance, Xue *et al.* found that feature selection methods are capable of reducing the noise generated by the use of overlapping and redundant molecular descriptors, and in some cases, improving the accuracy of SVM classification of pharmacokinetic behaviour of chemical agents [162]. The use of all available descriptors likely results in the inclusion of partially redundant information, some of which may to some extent become noise that interferes with the prediction results or obscures relevant information. In our study, for example, the three autocorrelation descriptor-sets (Sets D3, D4 and D5) all utilize the same physicochemical properties, only differing in the correlation algorithm. Amino acid composition information is also repeated in Sets D1, D7, D8 and D9. Based on the observation from this study as well as results of previous studies [148, 162, 269, 270], it

is possible that feature selection methods may be helpful in selecting the optimal set of descriptors to improve prediction accuracy in addition to computing efficiency for predicting protein functional families.

4 CONCLUSIONS AND FUTURE WORK

This last section summarizes the results of this work (Sec. 4.1) and its contribution towards the problem of protein functional prediction (Sec. 4.2). Caveats are noted (Sec. 4.3) and future directions (Sec. 4.4) are also suggested.

4.1 Findings

In this study, the efficacy of ten protein descriptor-sets in six protein functional family prediction using SVM was evaluated. Corroborating with previous work done on protein descriptors [135, 140, 180, 181, 203, 225, 271], it was found that the descriptor-sets evaluated in this work, which comprise some of the most commonly used descriptors, generally return good results and do not differ significantly. In particular, the use of combination descriptor-sets tends to give slightly better prediction performance than the use of individual descriptor-sets; moreover, the performance of pair-wise combination descriptor-sets were comparable to that of the entire combination of all descriptor-sets. This argues well for the use of a reduced descriptor-set. Lastly, descriptor-sets that utilize a combination of composition and sequence order or correlation information generally ranked well.

4.2 Contributions

The ramifications of this work are two-fold — in the fields of protein functional prediction and machine learning, both of which will continue to grow in importance, particularly at this juncture in current research. The availability of entire genome sequences and high-throughput facilities makes the tasks of assigning functions to novel proteins one of the most pressing problems in the post-genomic era [272]. At the same time, the use of machine learning systems will similarly become more widespread as the volume of biological data grows rapidly and the data analysis required becomes more complex, resulting in problems that cannot be solved by classical programming techniques and necessitating the utilization of techniques that can deal with such problem domains. The search for reliable methods for assigning protein function is of paramount importance, and not only on the side of the computational biologists; laboratory biologists themselves remain divided over accuracy of functional annotations of genomes [44–48]. To this end, SVM is considered to be one such method and has in fact shown to be a robust learner for noisy and complex domains because of two key features: good generalization capability and kernel functions. This work explored one crucial component of the SVM methodology — the representation of data.

4.3 Caveats

It should be noted that the performance of machine learning methods depends critically on a number of factors such as the quality of the training dataset (in particular example

diversity) and the machine learning method used. The datasets used in this work are not expected to be fully representative of all protein functional families; similarly, the descriptor-sets covered in this study comprise a limited subset of the descriptors available. Therefore, conclusions from this study might not be readily extended to other datasets or other descriptor-sets.

4.4 Future Directions

While there seems to be no preferred descriptor-set that could be utilized for all datasets as prediction results is highly dependent on datasets, the performance of protein classification may be enhanced by using established feature selection methods [271, 273]. Future work could look into the selection of optimal combinations of descriptors using such methods, or through combinations of the better performing descriptors.

Some of the descriptor-sets used in this work were highly similar, yet showed significantly different performances. For example, the combination sets D7, D8 and D9 all make use of only amino acid composition and sequence order information, yet there was a clear trend in predictive performance (D8 > D9 > D7). The most obvious difference between these sets lies in the way they represent sequence order information, thus, investigating the exact difference(s) and the reason for its effect(s) could further help to focus subsequent work in either improving existing descriptors or designing new descriptors. Alternatively, the incorporation of appropriate sets of physicochemical

properties not covered by some of the existing descriptor-sets could also help improve performance.

In conclusion, while current descriptors in machine learning methods have shown good results in protein functional prediction studies, there are still a lot of potential research areas that remains to be explored.

BIBLIOGRAPHY

1. Bork, P., et al., *Predicting function: from genes to genomes and back.* J Mol Biol, 1998. **283**: p. 707-725.
2. Eisenberg, D., et al., *Protein function in the post-genomic era.* Nature, 2000. **405**: p. 823-826.
3. Downward, J., *The ins and outs of signalling.* Nature, 2001. **411**: p. 759–762.
4. Draper, D.E., *Themes in RNA-protein recognition.* J Mol Biol, 1999. **293**: p. 255-270.
5. Lengeler, J.W., *Metabolic networks: A signal-oriented appaoch to cellular models.* Biol Chem, 2000. **381**: p. 911–920.
6. Siomi, H. and G. Dreyfuss, *RNA-binding proteins as regulators of gene expression.* Curr Opin Genet Dev, 1997. **7**: p. 345–353.
7. Cai, C.Z., et al., *SVM-Prot: Web-based support vector machine software for functional classification of a protein from its primary sequence.* Nuclei Acid Res, 2003. **31**: p. 3692-3697.
8. Cai, C.Z., et al., *Enzyme family classification by support vector machines.* Proteins, 2004. **55**: p. 66-76.
9. Karchin, R., K. Karplus, and D. Haussler, *Classifying G-protein coupled receptors with support vector machines.* Bioinformatics, 2002. **18**: p. 147-159.
10. Baldi, P. and G. Pollastri, *A machine-learning strategy for protein analysis* IEEE Intell Sys Biol, 2002. **17**(2): p. 28–35.
11. Hunkapiller, T., et al., *Large-scale and automated DNA sequence determination.* Science, 1991. **254**: p. 59-67.
12. Roberts, L., *Large-scale sequencing trials begin.* Science, 1990. **250**: p. 1336-1338.
13. Fleischmann, W., et al., *A novel method forautomatic functional annotation of proteins.* Bioinformatics, 1999. **15**: p. 228-233.
14. Holm, L. and C. Sander, *Protein folds and families: sequence and structure alignments.* Nuclei Acid Res, 1999. **27**: p. 244-247.
15. Luscombe, N., R. Laskowski, and J. Thornton, *Amino acid-base interactions: a three-dimensional analysis of protein-DNA interactions at anatomic level.* Nuclei Acid Res, 2001. **29**: p. 2860-2874.
16. Thornton, J., *From genome to function.* Science, 2000. **292**: p. 2095-2097.
17. Valencia, A., *Bioinformatics: biology by other means.* Bioinformatics, 2002. **18**: p. 1551-1552.
18. Valencia, A. and F. Pazos, *Computational methods for the prediction of protein interactions.* Curr Opin Struct Biol, 2002. **12**: p. 368-373.
19. Airozo, D., et al., *MEDLINE® (Medical Literature Analysis and Retrieval System Online)* 1999.
20. Rost, B., et al., *Automatic prediction of protein function.* Cell Mol Life Sci, 2003. **60**(12): p. 2637-2650.
21. Pevzner, A., *Computational Molecular Biology, An Algorithmic Approach*. 2000: The MIT Press.
22. Baldi, P. and S. Brunak, *Bioinformatics: The Machine Learning Approach*. 2nd ed. 2001: The MIT Press.
23. Wei, C., et al., *Closing in on the C. elegans ORFeome by cloning TWINSCAN predictions.* Genome Res, 2005. **15**: p. 577-582.
24. Smialowski, P., et al., *Predicting experimental properties of proteins from sequence by machine learning techniques.* Curr Protein Pept Sci, 2007. **8**(2): p. 121–133.
25. Pawlowski, K., et al. *Sensitive sequence comparison as protein function predictor.* in *Pacific Symposium on Biocomputing* 2000. Hawaii: World Scientific Publishing.

26. Ahmad, S., M. Gromiha, and A. Sarai, *Analysis and prediction of DNA-binding proteins and their binding residues based on composition, sequence and structural information.* Bioinformatics, 2004. **17**(11): p. 1027-1034.
27. Bhasin, M. and G.P. Raghava, *Classification of nuclear receptors based on amino acid composition and dipeptide composition.* J Biol Chem, 2004. **279**: p. 23262-23266.
28. des Jardin, M., et al., *Prediction of enzyme classification from protein sequence without the use of sequence similarity.* Proc Int Conf Intell Syst Mol Biol, 1997. **5**: p. 92-99.
29. Jensen, L.J., *Prediction of human protein function from post-translational modifications and localization features.* J Mol Biol, 2002. **319**: p. 1257-1265.
30. Altschul, S.F., et al., *Basic local alignment search tool.* J Mol Biol, 1990. **215**(3): p. 403-410.
31. *Twenty years of citation superstars*. Science Watch 2003 [cited 2007 April 17]; Available from: http://www.sciencewatch.com/sept-oct2003/sw_sept-oct2003_page1.htm.
32. Russo, E. and S. Bunk, *Hot papers in bioinformatics.* The Scientist, 1999. **13**(8): p. 15.
33. Altschul, S.F. and E.V. Koonin, *Iterated profile searches with PSI-BLAST--a tool for discovery in protein databases.* Trends Biochem Sci, 1998. **23**: p. 444-447.
34. Bork, P. and E.V. Koonin, *Protein sequence motifs.* Curr Opin Struct Biol, 1996. **6**: p. 366-376.
35. Kasuya, A. and J.M. Thornton, *Three-dimensional structure analysis of PROSITE patterns.* J Mol Biol, 1999. **286**: p. 1673-1691.
36. Hodges, H.C. and J.W. Tsai, *3D-Motifs: An informatics approach to protein function prediction.* FASEB J, 2002. **16**: p. A543.
37. Gattiker, A., E. Gasteiger, and A. Bairoch, *ScanProsite: a reference implementation of a PROSITE scanning tool.* Appl Bioinformatics, 2002. **1**: p. 107-108.
38. Benner, S.A., et al., *Functional inferences from reconstructed evolutionary biology involving rectified databases--an evolutionarily grounded approach to functional genomics.* Res Microbiol, 2000. **151**: p. 97-106.
39. Scott, K.A. and V. Daggett, *Folding mechanisms of proteins with high sequence identity but different folds.* Biochemistry, 2007. **46**(6): p. 1545-1556.
40. Blundell, T. and M. Johnson, *Catching a common fold.* Protein Sci, 1993. **2**(6): p. 877-883.
41. Sander, C. and R. Schneider, *Database of homology-derived protein structures and the structural meaning of sequence alignment.* Proteins, 1991. **9**: p. 56-68.
42. Enright, A.J. and C.A. Ouzounis, *GeneRAGE: a robust algorithm for sequence clustering and domain detection.* Bioinformatics, 2000. **16**: p. 451-457.
43. Wilson, C.A., J. Kreychman, and M. Gerstein, *Assessing annotation transfer for genomics: quantifying the relations between protein sequence, structure and function through traditional and probabilistic scores.* J Mol Biol, 2000. **297**(1): p. 233-249.
44. Casari, G., et al., *Challenging times for bioinformatics.* Nature, 1995. **376**: p. 647–648.
45. Kryrpides, N.C. and C. Ouzounis, *Whole-genome sequence annotation: 'Going wrong with confidence'.* Mol Microbiol, 1999. **32**: p. 886-887.
46. Ouzounis, C., et al., *Novelties from the complete genome of Mycoplasma genitalium.* Mol Microbiol, 1996. **20**(4): p. 898-900.
47. Devos, D. and A. Valencia, *Intrinsic errors in genome annotation.* Trends Genet, 2001. **17**(8): p. 429-431.
48. Brenner, S.E., *Errors in genome annotation.* Trends Genet, 1999. **15**: p. 132-133.
49. Chothia, C. and A.M. Lesk, *The relation between the divergence of sequence and structure in proteins.* EMBO J, 1986. **5**: p. 823-826.
50. Doolittle, R.F., *Similar amino acid sequences: chance or common ancestry?* Science, 1981. **214**: p. 149-159.

51. Doolittle, R.F., *Of URFs and ORFs: A Primer on How to Analyze Derived Amino Acid Sequences*. 1986, Mill Valley, CA, USA: University Science Books.
52. Zuckerkandl, E. and L. Pauling, *Evolutionary Divergence and Convergence in Proteins*, in *Evolving Genes and Proteins*, V. Bryson and H.J. Vogel, Editors. 1965, Academic Press: New York. p. 97-166.
53. Skolnick, J., J.S. Fetrow, and A. Kolinski, *Structural genomics and its importance for gene function analysis.* Nat Biotechnol, 2000. **18**: p. 283-287.
54. Holm, L. and C. Sander, *Mapping the protein universe.* Science, 1996. **273**: p. 595-603.
55. Brenner, S.E., et al., *Understanding protein structure: using scop for fold interpretation.* Methods Enzymol, 1996. **266**: p. 635–643.
56. Holm, L. and C. Sander, *Dali/FSSP classification of three-dimensional protein folds.* Nuclei Acid Res, 1996. **25**: p. 231–234.
57. Valencia, A., et al., *GPTase domains of Ras p21 oncogene protein and elongation factor Tu: analysis of three dimensional structures, sequence families and functional sites.* Proc. Natl Acad. Sci. USA, 1991. **88**: p. 5443–5447.
58. Rost, B., *Protein structures sustain evolutionary drift.* Folding Des, 1997. **2**: p. S19–S24.
59. Levitt, M. and M. Gerstein, *A unified statistical framework for sequence comparison and structure comparison.* Proc Natl Acad Sci USA, 1998. **95**(11): p. 5913–5920.
60. Andreeva, A., et al., *SCOP database in 2004: refinements integrate structure and sequence family data.* Nuclei Acid Res, 2004. **32**: p. D226-229.
61. Di Gennaro, J.A., et al., *Enhanced functional annotation of protein sequences via the use of structural descriptors.* J Struct Biol, 2001. **134**: p. 232-245.
62. Ivanciuc, O., et al., *Using property based sequence motifs and 3D modeling to determine structure and functional regions of proteins.* Curr Med Chem, 2004. **11**: p. 583-593.
63. Stark, A. and R.B. Russell, *Annotation in three dimensions. PINTS: Patterns in Non-homologous Tertiary Structures.* Nuclei Acid Res, 2003. **31**: p. 3341-3344.
64. Wallace, A.C., N. Borkakoti, and J.M. Thornton, *TESS: a geometric hashing algorithm for deriving 3D coordinate templates for searching structural databases. Application to enzyme active sites.* Protein Sci, 1997. **6**: p. 2308-2323.
65. Brenner, S.E., *Target selection for structural genomics.* Nat Struct Biol, 2000. **7**(Suppl): p. 967-969.
66. Teichmann, S.A., C. Chothia, and M. Gerstein, *Advances in structural genomics.* Curr. Opin. Struct. Biol., 1999. **9**: p. 390-399.
67. Bonneau, R. and D. Baker, *Ab initio protein structure prediction: progress and prospects.* Annual Review of Biophysics and Biomolecular Structure, 2001. **30**: p. 173–189.
68. Bartlett, G.J., N. Borkakoti, and J.M. Thornton, *Catalysing new reactions during evolution: economy of residues and mechanism.* J Mol Biol, 2003. **331**: p. 829-860.
69. Orengo, C.A., A.E. Todd, and J.M. Thornton, *From protein structure to function.* Curr Opin Struct Biol, 1999. **9**: p. 374-382.
70. Shakhnovich, B.E., et al., *Functional fingerprints of folds: evidence for correlated structure-function evolution.* J Mol Biol, 2003. **326**: p. 1-9.
71. Todd, A.E., C.A. Orengo, and J.M. Thornton, *Evolution of function in protein superfamilies, from a structural perspective.* J Mol Biol, 2001. **307**: p. 1113-1143.
72. Henikoff, S., et al., *Gene families: the taxonomy of protein paralogs and chimeras.* Science, 1997. **278**: p. 609-614.
73. Lipman, D.J. and W.R. Pearson, *Rapid and sensitive protein similarity searches.* Science, 1985. **227**: p. 1435-1441.
74. Holm, L. and C. Sander, *Structural alignment of globins, phycocyanins and colicin A.* FEBS Lett, 1993. **315**: p. 301-306.
75. Russell, S.J. and P. Norvig, *Artificial Intelligence: A Modern Approach*. 2003, Upper Saddle River, NJ, Great Britain: Prentice Hall.

76. Shortle, D., et al., *Protein folding for realists: a timeless phenomenon.* Prot Sci, 1996. **5**: p. 991–1000.
77. van Gunsteren, W.F., *Molecular dynamics studies of proteins.* Curr Opin Struct Biol, 1993. **3**: p. 167–174.
78. Wang, L.H., J. Liu, and H.B. Zhou. *A comparison of two machine learning methods for protein secondary structure prediction*. in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*. 2004. Shanghai, China.
79. Mitchell, T.M., *Machine Learning*. 1997, New York: McGraw-Hill.
80. Michie, D., D.J. Spiegelhalter, and C.C. Taylor, *Machine Learning, Neural and Statistical Classification*. 1994, London: Ellis Horwood.
81. Nilsson, N.J., *Introduction to Machine Learning, Draft of Incomplete Notes*. 2005.
82. Quinlan, J.R., *Induction of decision trees.* Mach Learn, 1986. **1**(1): p. 81-106.
83. Jain, A.K., R.P.W. Duin, and J. Mao, *Statistical pattern recognition: a review.* IEEE Trans Pattern Anal, 2000. **22**(1): p. 4–37.
84. ICML. *Workshop on Learning from Imbalanced Datasets II*. in *20th International Conference on Machine Learning*. 2003. Washington, DC.
85. Provost, F. *Learning with imbalanced data sets 101*. in *AAAI'2000 Workshop on Imbalanced Data Sets*. 2000. Austin, Texas.
86. Provost, F. and T. Fawcett, *Robust classification for imprecise environments.* Mach Learn, 2001. **42**: p. 203–231.
87. Müller, K.R., et al., *An introduction to kernel-based learning algorithms.* IEEE Trans Neural Net, 2001. **12**(2): p. 181–201.
88. Müller, K.R. and G. Orr, eds. *Neural Networks: Tricks of the Trade*. Vol. 1524. 1998, Springer LNCS.
89. Poggio, T. and F. Girosi, *Regularization algorithms for learning that are equivalent to multilayer networks.* Science, 1990. **247**: p. 978–982.
90. Breiman, L., *Bagging predictors.* Mach Learn, 1996. **24**(2): p. 123–140.
91. Chou, P.A., *Optimal partitioning for classification and regression trees.* IEEE Trans Pattern Anal 1991. **13**(4): p. 340–321.
92. Quinlan, J.R., *C4.5: Programs for Machine Learning*. 1993, San Mateo, California: Morgan Kaufmann.
93. Winston, P.H., *Artificial Intelligence*. 3rd ed. 1992: Addison–Wesley.
94. Cover, T. and P. Hart, *Nearest neighbor pattern classification.* IEEE Trans Inform Theor, 1967. **13**(1): p. 21–27.
95. Fix, E. and J.L. Hodges, *Discriminatory analysis—nonparametric discrimination: Consistency properties, Technical Report 21-49-004*. 1951, USAF School of Aviation Medicine, Randolph Field: Texas.
96. Cost, S. and S. Salzberg, *A weighted nearest neighbor algorithm for learning with symbolic features.* Mach Learn, 1993. **10**(1): p. 57–78.
97. Aha, D. and D. Kibler. *Noise-tolerant instance-based learning algorithms*. in *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*. 1989. Detroit, MI: Morgan Kaufmann.
98. Salzberg, S. *Nested hyper-rectangles for exemplar-based learning*. in *Analogical and Inductive Inference: International Workshop AII '89*. 1989. Berlin: Springer–Verlag.
99. Dasarathy, B.V., *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. 1990, Los Alamitos: IEEE Computer Society Press.
100. Cabello, D., et al., *Fuzzy K-nearest neighbor classifiers for ventricular arrhythmia detection.* Int J Biomed Comput, 1991. **27**: p. 77–93.
101. Hoffman, B., et al., *Quantitative structure-activity relationship modeling of dopamine D(1) antagonists using comparative molecular field analysis, genetic algorithms-partial*

least-squares, and K nearest neighbor methods. J Med Chem, 1999. **42**(17): p. 3217–3226.
102. Kauffman, G.W. and P.C. Jurs, *2001QSAR and k-nearest neighbor classification analysis of selective cyclooxygenase-2 inhibitors using topologically-based numerical descriptors.* J Chem Inf Comput Sci, 2001. **41**: p. 1553–1560.
103. Shen, M., et al., *Quantitative structure-activity relationship analysis of functionalized amino acid anticonvulsant agents using k nearest neighbor and simulated annealing PLS methods.* J Med Chem, 2002. **45**(13): p. 2811–2823.
104. Vouros, G.A. and T. Panayiotopoulos. *SETN 2004, LNAI 3025*. 2004. Heidelberg, Berlin: Springer–Verlag.
105. Hassoun, M.H., *Fundamentals of Artificial Neural Networks*. 1995, Cambridge: MIT Press.
106. Rosenblatt, F., *The Perceptron: A probabilistic model for information storage and organization in the brain.* Psychol Rev, 1985. **65**: p. 386–408.
107. Vapnik, V.N., *The Nature of Statistical Learning Theory*. 2000: Springer.
108. Winkler, D.A., *Neural networks as robust tools in drug lead discovery and development.* Mol Biotechnol, 2004. **27**: p. 139–168.
109. Cheng, J. and P. Baldi, *Three-stage prediction of protein {beta}-sheets by neural networks, alignments and graph algorithms.* Bioinformatics, 2005. **21**(Suppl 1): p. i75–i84.
110. Yang, Z.R., *Prediction of caspase cleavage sites using Bayesian bio-basis function neural networks.* J Bioinform Comput Biol, 2005. **2**: p. 511–531.
111. Aleksander, I. and H. Morton, *An Introduction to Neural Computing*. 2nd ed. 1995, London: International Thomson Computer Press.
112. Mooney, R., et al. *An experimental comparison of symbolic and connectionist learning algorithms*. in *Proceedings of the International Joint Conference on Artificial Intelligence*. 1989. San Mateo, CA: Morgan Kaufmann.
113. Shavlik, J., R. Mooney, and G. Towell, *Symbolic and neural learning algorithms: An experimental comparison (Technical Report #857)*. 1989, Computer Sciences Department, University of Wisconsin: Madison, WI.
114. Weiss, S. and I. Kapouleas. *An empirical comparison of pattern recognition, neural nets, and machine learning classification methods*. in *Proceedings of the International Joint Conference on Artificial Intelligence*. 1989. San Mateo, CA: Morgan Kaufmann.
115. Draghici, S. and R.B. Potter, *Predicting HIV drug resistance with neural networks.* Bioinformatics, 2003. **19**: p. 98–107.
116. Specht, D.F., *Probabilistic neural networks.* Neural Networks, 1990. **3**: p. 109–118.
117. Cortes, C. and V.N. Vapnik, *Support vector networks.* Mach Learn, 1995. **20**: p. 273–297.
118. Joachims, T. *A statistical learning model of text classification with support vector machines*. in *Proceedings of the 24th Conference on Research and Development in Information Retrevial (SIGIR)*. 2001. New Orleans: Association for Computing Machinery.
119. DeCoste, D. and B. Scholkopf, *Training invariant support vector machines.* Mach Learn, 2002. **46**(1): p. 161–190.
120. Fritsche, H.A., *Tumor markers and pattern recognition analysis: A new diagnostic tool for cancer.* J Clin Ligand Assay, 2002. **25**: p. 11–15.
121. Brown, M.P., et al., *Knowledge-based analysis of microarray gene expression data by using support vector machines.* Proc Natl Acad Sci USA, 2000. **97**(1): p. 262-267.
122. Hua, S. and Z. Sun, *A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach.* J Mol Biol, 2001. **308**: p. 397–407.

123. Bock, J.R. and D.A. Gough, *Predicting protein--protein interactions from primary structure.* Bioinformatics, 2001. **17**: p. 455-460.
124. Burges, C.J.C., *A tutorial on support vector machines for pattern recognition* Data Mining and Knowledge Discovery, 1998. **2**(2): p. 121–167.
125. van der Walt, C.M. and E. Barnard. *Data characteristics that determine classifier performance*. in *Proceedings of the Sixteenth Annual Symposium of the Pattern Recognition Association of South Africa*. 2005. Langebaan, South Africa.
126. Vapnik, V., *The Nature of Statistical Learning Theory*. 1995, Berlin: Springer-Verlag.
127. Vapnik, V., *Statistical Learning Theory*. 1998, New York: Wiley.
128. Burges, C.J.C. and D.J. Crisp. *Uniqueness of the SVM solution*. in *Neural Information Processing Systems 2000*. 2000. Vancouver, Canada.
129. Fletcher, R., *Practical Methods of Optimization*. 2nd ed. 1987: John Wiley and Sons, Inc.
130. Cristianini, N. and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*. 2000: Cambridge University Press.
131. Schölkopf, B. and e. al. *Prior knowledge in support vector kernels*. in *Advances in Neural Information Processing Systems*. 1998. Denver, CO, USA: MIT Press.
132. Burges, C.J.C. *Simplified support vector decision rules*. in *Proceedings of the 13th International Conference on Machine Learning*. 1996. Bari, Italy: Morgan Kaufmann.
133. Joachims, T., *Text categorization with Support Vector Machines. Technical Report, LS VIII Number 23*. 1997, University of Dotmund.
134. Boser, B.E., I.M. Guyon, and V. Vapnik. *A training algorithm for optimal margin classifiers*. in *Fifth Annual Workshop on Computational Learning Theory*. 1992. Pittsburg: ACM.
135. Han, L.Y., et al., *Predicting functional family of novel enzymes irrespective of sequence similarity: a statistical learning approach.* Nuclei Acid Res, 2004. **32**(21): p. 6437-6444.
136. Li, Z.R., et al., *PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence.* Nuclei Acid Res, 2006. **34**(Web Server issue): p. W32-W37.
137. Lo, S.L., et al., *Effect of training datasets on support vector machine prediction of protein-protein interactions.* Proteomics, 2005. **5**: p. 876-884.
138. Cui, J., et al., *Prediction of MHC-binding peptides of flexible lengths from sequence-derived structural and physicochemical properties.* Mol Immunol, 2006. **44**(5): p. 866-877.
139. Dubchak, I., et al., *Recognition of a protein fold in the context of the Structural Classification of Proteins (SCOP) classification.* Proteins, 1999. **35**: p. 401-407.
140. Han, L.Y., et al., *Prediction of RNA-binding proteins from primary sequence by a support vector machine approach.* RNA, 2004. **10**: p. 355-368.
141. Bock, J.R. and D.A. Gough, *Whole-proteome interaction mining.* Bioinformatics, 2003. **19**: p. 125-134.
142. Chou, K.C., *Prediction of protein subcellular locations by incorporating quasi-sequence-order effect.* Biochem Biophys Res Commun, 2000. **278**(2): p. 477-483.
143. Chou, K.C. and Y.D. Cai, *Prediction of protein subcellular locations by GO-FunD-PseAA predictor.* Biochem Biophys Res Commun, 2004. **320**(4): p. 1236-1239.
144. Schneider, G. and P. Wrede, *The rational design of amino acid sequences by artificial neural networks and simulated molecular evolution: de novo design of an idealized leader peptidase cleavage site.* Biophys J, 1994. **66**: p. 335-344.
145. Farnum, M., R. DesJarlais, and D.K. Agrafiotis, *Molecular diversity*, in *Handbook of Chemoinformatics: From Data to Knowledge*, J. Gasteiger, Editor. 2003, Wiley: Chichester. p. 1641-1685.
146. Todeschini, R. and V. Consonni, *Handbook of Molecular Descriptors*. 2000, Weinheim: Wiley.

147. Zhang, Z.D., S. Kochhar, and M.G. Grigorov, *Descriptor-based protein remote homology identification.* Protein Sci, 2005. **14**: p. 431-444.
148. Al-Shahib, A. and R.D.G. Breitling, *Feature selection and the class imbalance problem in predicting protein function from sequence.* Appl Bioinformatics, 2005. **4**(3): p. 195-203.
149. Over, T.M., *The best two independent measurements are not the two best.* IEEE Trans Syst Man Cyb, 1965. **14**: p. 326–334.
150. Jain, A.K. and B. Chandrasekaran, *Dimensionality and sample size considerations in pattern recognition practice*, in *Handbook of Statistics*, P.R. Krishnaiah and I.N. Kanal, Editors. 1982, North-Holland: Amsterdam. p. 835–855.
151. Watanabe, S., *Pattern Recognition: Human and Mechanical*. 1985, New York: Wiley.
152. Guyon, I. and A. Elisseeff, *An introduction to variable and feature selection.* J Machine Learn Res, 2003. **3**: p. 1157-1182.
153. Reeves, S.J. *An improved sequential backward selection algorithm for large-scaleobservation selection problems*. in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 1998. Seattle, Washington, USA.
154. Brown, R. and Y. Martin, *Use of structure-activity data to compare structure-based clustering methods and descriptors for use in compound selection.* J Chem Inf Comput Sci, 1996. **36**(3): p. 572-584.
155. Cramer, R.D., D.E. Patterson, and J. Bunce, *Comparative molecular field analysis (CoMFA): 1. Effect of shape on binding of steroids to carrier proteins.* J Am Chem Soc, 1988. **110**: p. 5959-5967.
156. Glen, W., W. Dunn, and R. Scott, *Principal components analysis and partial least squares regression.* Tetrahedron Comput Methodol, 1989. **2**: p. 349-376.
157. Matter, H., *Selecting optimally diverse compounds from structure databases a validation study of two-dimensional and three-dimensional molecular descriptors.* J Med Chem, 1997. **40**(8): p. 1219-1229.
158. Matter, H. and T. Pötter, *Comparing 3D pharmacophore triplets and 2D fingerprints. for selecting diverse compound subsets.* J Chem Inf Comput Sci, 1999. **39**: p. 1211-1225.
159. Patterson, D.E.P., et al., *Neighborhood behavior: A useful concept for validation of "molecular diversity" descriptors.* J Med Chem, 1996. **39**(16): p. 3049-3059.
160. Xue, L. and J. Bajorath, *Molecular descriptors in chemoinformatics, computational combinatorial chemistry, and virtual screening.* Comb Chem High Throughput Screen, 2000. **3**(5): p. 363-372.
161. Xue, L., J. Godden, and J. Bajorath, *Identification of a preferred set of descriptors for compound classification based on principcal component analysis.* J Chem Inf Comput Sci, 1999. **39**: p. 669-704.
162. Xue, L., J. Godden, and J. Bajorath, *Evaluation of descriptors and mini-fingerprints for the identification of molecules with similar activity.* J Chem Inf Comput Sci, 2000. **40**(5): p. 1227-1234.
163. de Cerqueira Lima, P., et al., *Combinatorial QSAR modeling of P-glycoprotein substrates.* J Chem Inf Model, 2006. **46**(3): p. 1245-1254.
164. Katritzky, A. and E. Gordeeva, *Traditional topological indices vs electronic, geometrical, and combined molecular descriptors in QSAR/QSPR research.* J Chem Inf Comput Sci, 1993. **33**(6): p. 835-857.
165. Kovatcheva, A., et al., *Combinatorial QSAR of ambergris fragrance compounds.* J Chem Inf Comput Sci, 2004. **44**(2): p. 582-595.
166. Burbidge, R., et al., *Drug design by machine learning: support vector machines for pharmaceutical data analysis.* Comput Chem, 2001. **26**(1): p. 5-14.

167. Platt, J.C., *Sequential Minimal Optimization: A fast algorithm for training support vector machines*, in *Microsoft Research. Technical Report MSR-TR-98-14*. 1998.
168. Osuna, E., R. Freund, and F. Girosi. *An improved training algorithm for support vector machines*. in *Neural Networks for Signal Processing VII-Proceedings of the 1997 IEEE Workshop*. 1997. Amelia Island, FL, USA.
169. Aizerman, M., E. Braverman, and L. Rozonoer, *Theoretical foundations of the potential function method in pattern recognition learning.* Automat Rem Contr 1964. **25**: p. 821–837.
170. Courant, R. and D. Hilbert, *Methods of Mathematical Physics*. 1953: Interscience.
171. Schölkopf, B., et al., *Comparing support vector machines with gaussian kernels to radial basis function classifiers.* IEEE Trans Sign Process, 1997. **45**: p. 2758–2765.
172. Chou, K.C., *Prediction of membrane protein types by incorporating amphipathic effects.* J Chem Inf Model, 2005. **45**(2): p. 407-413.
173. Gao, Q.B., et al., *Prediction of protein subcellular location using a combined feature of sequence.* FEBS Lett, 2005. **579**(16): p. 3444-3448.
174. Broto, P., G. Moreau, and C. Vandicke, *Molecular structures: perception, autocorrelation descriptor and SAR studies.* Eur J Med Chem, 1984. **19**: p. 71-78.
175. Moreau, G. and P. Broto, *Autocorrelation of molecular structures, application to SAR studies.* Nour J Chim, 1980. **4**: p. 757-764.
176. Moran, P.A., *Notes on continuous stochastic phenomena.* Biometrika, 1950. **37**: p. 17-23.
177. Geary, R.C., *The contiguity ratio and statistical mapping.* The Incorporated Statistician, 1954. **5**: p. 115-145.
178. Dubchak, I., et al., *Prediction of protein folding class using global description of amino acid sequence.* Proc Natl Acad Sci USA, 1995. **92**: p. 8700-8704.
179. Grantham, R., *Amino acid difference formula to help explain protein evolution.* Science, 1974. **185**: p. 862-864.
180. Chou, K.C., *Prediction of protein cellular attributes using pseudo-amino acid composition.* Proteins 2001. **43**(3): p. 246-255.
181. Lin, H.H., et al., *Prediction of transporter family from protein sequence by support vector machine approach.* Proteins, 2006. **62**: p. 218-231.
182. Shepherd, A.J., D. Gorse, and J.M. Thornton, *A novel approach to the recognition of protein architecture from sequence using Fourier analysis and neural networks.* Proteins, 2003. **50**(2): p. 290-302.
183. Eisenhaber, F., et al., *Prediction of secondary structural content of proteins from their amino acid composition alone. I. New analytic vector decomposition methods.* Proteins, 1996. **25**(2): p. 157-168.
184. Grassmann, J., et al., *Protein fold class prediction: new methods of statistical classification.* Proc Int Conf Intell Syst Mol Biol, 1999: p. 106-112.
185. Reczko, M. and H. Bohr, *The DEF data base of sequence based protein fold class predictions.* Nuclei Acid Res, 1994. **22**(17): p. 3616-3619.
186. Chou, K.C. and Y.D. Cai, *Using functional domain composition and support vector machines for prediction of protein subcellular location.* J Biol Chem, 2002. **277**: p. 45765-45769.
187. Hua, S. and Z. Sun, *Support vector machine approach for protein subcellular localization prediction.* Bioinformatics, 2001. **17**(8): p. 721-728.
188. Kawashima, S. and M. Kanehisa, *AAindex: amino acid index database.* Nuclei Acid Res, 2000. **28**(1): p. 374.
189. Cid, H., et al., *Hydrophobicity and structural classes in proteins.* Protein Eng, 1992. **5**(5): p. 373-375.
190. Bhaskaran, R. and P.K. Ponnuswammy, *Positional flexibilities of amino acid residues in globular proteins.* Int J Pept Protein Res, 1988. **32**: p. 242-255.

191. Charton, M. and B.I. Charton, *The structural dependence of amino acid hydrophobicity parameters.* J Theor Biol, 1982. **99**: p. 626-644.
192. Chothia, C., *The nature of the accessible and buried surfaces in proteins.* J Mol Biol, 1976. **105**: p. 1-12.
193. Bigelow, C.C., *On the average hydrophobicity of proteins and the relation between it and protein structure.* J Theor Biol, 1967. **16**: p. 187-211.
194. Charton, M., *Protein folding and the genetic code: an alternative quantitative model.* J Theor Biol, 1981. **91**: p. 115-123.
195. Dayhoff, H. and H. Calderone, *Composition of proteins.* Altas Protein Seq Struct, 1978. **5**: p. 363-373.
196. Japan, G., *Amino acid indices and similarity matrices*. 2005.
197. Lin, Z. and X.M. Pan, *Accurate prediction of protein secondary structural content.* J Protein Chem, 2001. **20**(3): p. 217-220.
198. Home, D.S., *Prediction of protein helix content from an autocorrelation analysis of sequence hydrophobicities.* Biopolymers, 1988. **27**: p. 451–477.
199. Sokal, R.R. and B.A. Thomson, *Population structure inferred by local spatial autocorrelation: an example from an Amerindian tribal population.* Am J Phys Anthropol, 2006. **129**: p. 121–131.
200. Tomii, K. and M. Kanehisa, *Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins.* Protein Eng, 1996. **9**: p. 27-36.
201. Damborsky, J., *Quantitative structure-function and structure-stability relationships of purposely modified proteins.* Protein Eng, 1998. **11**(1): p. 21-30.
202. Hopp, T.P. and K.R. Woods, *Prediction of protein antigenic determinants from amino acid sequences.* Proc Natl Acad Sci USA, 1981. **78**(6): p. 3824-3828.
203. Chou, K.C., *Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes.* Bioinformatics, 2005. **21**: p. 10-19.
204. Feng, Z.P., *An overview on predicting the subcellular location of a protein.* In Silico Biol, 2002. **2**: p. 291-303.
205. Wu, C.H., et al., *Protein family classification and functional annotation.* Comput Biol Chem, 2003. **27**: p. 37.
206. Bateman, A., et al., *The Pfam protein families database.* Nuclei Acid Res, 2004. **32**(Database issue): p. D138-D141.
207. Corpet, F., et al., *ProDom and ProDom-CG: Tools for protein domain analysis and whole-genome comparisons.* Nuclei Acid Res, 2000. **28**: p. 267–269.
208. Barker, W.C., F. Pfeiffer, and D.G. George, *Superfamily classification in PIR international protein sequence database.* Methods Enzymol, 1996. **266**: p. 59–71.
209. Yona, G., N. Linial, and M. Linial, *ProtoMap: Automatic classification of protein sequences and hierarchy of protein families.* Nuclei Acid Res, 2000. **28**: p. 49–55.
210. Falquet, L., et al., *The PROSITE database, its status in 2002.* Nuclei Acid Res, 2002. **30**(1): p. 235–238.
211. Attwood, T.K., et al., *PRINTS and PRINTS-S shed light on protein ancestry.* Nuclei Acid Res, 2002. **30**: p. 239–241.
212. Lo Conte, L., et al., *SCOP database in 2002: Refinements accommodate structural genomics.* Nuclei Acid Res, 2002. **30**: p. 264–267.
213. Pearl, F.M.G., et al., *A rapid classification protocol for the CATH domain database to support structural genomics.* Nuclei Acid Res, 2001. **29**: p. 223–227.
214. Huang, H., C. Xiao, and C.H. Wu, *ProClass protein family database.* Nuclei Acid Res, 2000. **28**: p. 273–276.
215. Wu, C.H., et al., *iProClass: An integrated, comprehensive, and annotated protein classification database.* Nuclei Acid Res, 2001. **29**: p. 52–54.

216. Apweiler, R., et al., *The InterProt Database, an integrated documentation resource for protein families, domains, and functional sites.* Nuclei Acid Res, 2001. **29**: p. 37–40.
217. Chou, K.C. and Y.D. Cai, *Predicting enzyme family class in a hybridization space.* Protein Sci, 2004. **13**: p. 2857-2863.
218. Chou, K.C. and D.W. Elrod, *Prediction of enzyme family classes.* J Proteome Res, 2003. **2**: p. 183-190.
219. NC-IUBMB, *Enzyme Nomenclature*. 1992, San Diego, California: Academic Press.
220. Chou, K.C., *Prediction of G-protein-coupled receptor classes.* J Proteome Res, 2005. **4**: p. 1413-1418.
221. Chou, K.C. and D.W. Elrod, *Bioinformatical analysis of G-protein-coupled receptors.* J Proteome Res, 2002. **1**: p. 429-433.
222. Busch, W. and M.H.J. Saier, *The transporter classification (TC) system.* Crit Rev Biochem Mol Biol, 2002. **37**(5): p. 287-337.
223. TCDB, *Transport Classification Database*, Saier Lab Bioinformatics Group.
224. Suzuki, J.Y., D.W. Bollivar, and C.E. Bauer, *Genetic analysis of chlorophyll biosynthesis.* Ann Rev Genet, 1997. **31**: p. 61-89.
225. Lin, H.H., et al., *Prediction of the functional class of lipid binding proteins from sequence-derived properties irrespective of sequence similarity.* J Lipid Res, 2006. **47**: p. 827-831.
226. Dutta, A.S. and A. Garner, *The pharaceutical industry and research in 2002 and beyond.* Drug News Perspect, 2003. **16**(10): p. 637–648.
227. Joet, T., et al., *Why is the plasmodium falciparum hexose transporter a promising new drug target?* Expert Opin Ther Target, 2003. **7**(5): p. 593-602.
228. Baenzigner, J.U., *Protein-specific glycosyltransferase: how and why they do it!* FASEB J, 1994. **8**(13): p. 1019-1025.
229. Kapitonov, D. and R.K. Yu, *Conserved domains of glycosyltransferase.* Glycobiology, 1999. **9**: p. 961-978.
230. Drews, J., *Genomic sciences and the medicine of tomorrow.* Nat Biotechnol, 1996. **14**(11): p. 1516-1518.
231. Gudermann, T.B., B. Nurnberg, and G. Schultz, *Receptors and G proteins as primary components of transmembrane signal transduction. Part 1. G-protein-coupled receptors: structure and function.* J MOl Med, 1995. **73**(2): p. 51-63.
232. Muller, G., *Towards 3D structures of G protein-coupled receptors: a multidisciplinary approach.* Curr Med Chem, 2000. **7**(9): p. 861-888.
233. Paulson, J.C. and K.J. Colley, *Glycosyltransferase.* J Biol Chem, 1989. **264**(30): p. 17645-17618.
234. Filmore, D., *It's a GPCR world*, in *Modern Drug Discovery (American Chemical Society)*. 2004.
235. Borst, P. and R.O. Elferink, *Mammalian ABC transporters in health and disease.* Ann Rev Biochem, 2002. **71**: p. 537-592.
236. Hediger, M.A., *Structure, function and evolution of solute transporters in prokaryotes and eukaryotes.* J Exp Biol, 1994. **196**: p. 15-49.
237. Seal, R.P. and S.G. Amara, *Excitatory amino acid transporters: a family in flux.* Ann Rev Pharmacol Toxicol, 1999. **39**: p. 431-456.
238. Saier, M.H.J., *A functional-phylogenetic classification system for transmembrane solute transporters.* Microbiol Mol Biol Rev, 2000. **64**: p. 351-411.
239. Beale, S.I. and J.D. Weinstein, *Biochemistry and regulation of photosynthetic pigment formation in plants and algae*, in *Biosynthesis of Tetrapyrroles*, P. Jordan, Editor. 1991, Elsevier: Amsterdam. p. 155-235.
240. Glatz, J.F., et al., *Cellular lipid binding proteins as facilitators and regulators of lipid metabolism.* Mol Cell Biochem, 2002. **239**: p. 3-7.

241. Downes, C.P., A. Gray, and J.M. Lucocq, *Probing phosphoinositide functions in signaling and membrane trafficking.* Trends Cell Biol, 2005. **15**: p. 259-268.
242. Bernlohr, D.A., et al., *Intracellular lipid-binding proteins and their genes.* Ann Rev Nutr, 1997. **17**: p. 277-303.
243. Niggli, V., *Structural properties of lipid-binding sites in cytoskeletal proteins.* Trends Biochem Sci, 2001. **26**: p. 604-611.
244. Balla, T., *Inositol-lipid binding motifs: signal integrators through protein-lipid and protein-protein interactions.* J Cell Sci, 2005. **118**: p. 2093-2104.
245. Pebay-Peyroula, E. and J.P. Rosenbusch, *High-resolution structures and dynamics of membrane protein--lipid complexes: a critique.* Curr Opin Struct Biol, 2001. **11**: p. 427-432.
246. Palsdottir, H. and C. Hunte, *Lipids in membrane protein structures.* Biochim Biophys Acta, 2004. **1666**: p. 2-18.
247. Burd, C.G. and G. Dreyfuss, *Conserved structures and diversity of functions of RNA-binding proteins.* Science, 1994. **265**: p. 615-621.
248. Kiledjian, M., et al., *Structure and function of hnRNP proteins*, in *RNA-Protein Interactions: Frontiers in Molecular Biology*, K. Nagai and I. Mattaj, Editors. 1994, IRL Press: Oxford. p. 127-149.
249. Fierro-Monti, I. and M.B. Mathews, *Proteins binding to duplexed RNA: one motif, multiple functions.* Trends Biochem Sci, 2000. **25**: p. 241-246.
250. Perculis, B.A., *RNA-binding proteins: if it looks like a sn(o)RNA.* Curr Biol, 2000. **10**: p. R916-R918.
251. Perez-Canadillas, J.M. and G. Varani, *Recent advances in RNA-protein recognition.* Curr Opin Struct Biol, 2001. **11**: p. 53-58.
252. Frank, D.N. and N.R. Pace, *Ribonuclease P: Unity and diversity in a tRNA processing ribozyme.* Ann Rev Biochem, 1998. **67**: p. 153–180.
253. Cesari, G., C. Sander, and A. Valencia, *A method to predict functional residues in proteins.* Nat Struct Biol, 1995. **2**: p. 171–178.
254. Elcock, A.H. and J.A. McCammon, *Calculation of weak protein-protein interactions: The pH dependence of the second virial coefficient.* Biophysical, 2001. **80**: p. 613–625.
255. Pawson, T., *Protein molecules and signaling networks.* Nature, 1995. **373**: p. 573–580.
256. Hermann, T. and E. Westhof, *Simulations of the dynamics at an RNA-protein interface.* Nat Struct Biol, 1999. **6**: p. 540–544.
257. Boeckmann, B., et al., *The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003.* Nuclei Acid Res, 2003. **31**(1): p. 365-370.
258. Saier, M.H.J., C.V. Tran, and R.D. Barabote, *TCDB: the Transporter Classification Database for membrane transport protein analyses and information.* Nuclei Acid Res, 2006. **34**(Database issue): p. D181-D186.
259. Heyer, L.J., S. Kruglyak, and S. Yooseph, *Exploring expression data: identification and analysis of coexpressed genes.* Genome Res, 1999. **9**(11): p. 1106-1115.
260. Li, W.Z. and A. Godzik, *Cd-hit: a fast program for clustering and comparing large sets of proteins or nucleotide sequences.* Bioinformatics, 2006. **22**: p. 1658-1659.
261. Li, W.Z., L. Jaroszewksi, and A. Godzik, *Clustering of highly homologous sequences to reduce the size of large protein database.* Bioinformatics, 2001. **17**: p. 282-283.
262. Li, W.Z., L. Jaroszewksi, and A. Godzik, *Tolerating some redundancy significantly speeds up clustering of large protein databases.* Bioinformatics, 2002. **18**: p. 77-82.
263. Baldi, P., et al., *Assessing the accuracy of prediction algorithms for classification: an overview.* Bioinformatics, 2000. **16**(5): p. 412-424.
264. Provost, F., T. Fawcett, and R. Kohavi. *The case against accuracy estimation for comparing induction algorithms*. in *Proc 15th International Conf on Machine Learning*. 1998. San Francisco, California: Morgan Kaufmann.

265. Veropoulos, K., C. Campbell, and N. Cristianini. *Controlling the sensitivity of support vector machines*. in *Proceedings of the International Joint Conference on Artificial Intelligence (UCAI99)*. 1999. Sweden: Morgan Kaufmann.
266. Kim, H. and H. Park, *Prediction of protein relative solvent accessibility with support vector machines and long-range interaction 3D local descriptor.* Proteins, 2004. **54**: p. 557-562.
267. Chinnasamy, A., W.K. Sung, and A. Mittal, eds. *Protein structure and fold prediction using tree-augmented bayesian classifier*. Pacific Symposium on Biocomputing, ed. R.B. Altman, et al. 2004, World Scientific Hawaii, USA. 387–398.
268. Dubchak, I., et al., *Prediction of protein folding class using global description of amino acid sequence.* Proc Natl Acad Sci USA, 1995. **92**(19): p. 8700–8704.
269. Al-Shahib, A., R.D.G. Breitling, and D. Gilbert, *FrankSum: new feature selection method for protein function prediction.* Int J Neural Syst, 2005. **15**: p. 259–275.
270. Xue, Y., et al., *Effect of molecular descriptor feature sleection in support vector machine classification of pharmacokinetic and toxicological properties of chemical agents.* J Chem Inf Comput Sci, 2004. **44**: p. 1630–1638.
271. Chen, C., et al., *Predicting protein structural class with pseudo-amino acid composition and support vector machine fusion network.* Anal Biochem, 2006. **357**: p. 116–121.
272. Hodgman, T.C., *A historical perspective on gene/protein funcitonal assignment.* Bioinformatics, 2000. **16**: p. 10–15.
273. Yu, H., et al. *Discovering compact and highly discriminative features or feature combinations of drug activities using support vector machines*. in *Proceedings of the IEEE Computer Society Bioinformatics Conference (CSB): 2003*. 2003. Standford, CA.