MILP Based Hyper-Box Enclosure Approach to Multi-Class Data Classification

by

Fadime Üney Yüksektepe

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Doctor of Philosophy

in

Industrial Engineering and Operations Management

Koç University

May 2009

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Fadime Üney Yüksektepe

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by the final

examining committee have been made

Committee Members:

Metin Türkay, Ph. D. (Advisor)

Kuban Altınel, Ph. D.

Ceyda Oğuz, Ph. D.

Serpil Sayın, Ph. D.

Deniz Yüret, Ph. D.

Date: 28.05.2009

I dedicate this thesis to

my husband, Gökhan, and my parents

for their constant support and unconditional love.

ABSTRACT

Data classification is an important data mining problem that aims to determine the membership of different instances to a number of different sets. Traditional approaches that are based on partitioning the data sets into two groups need some modifications for multi-class data classification problems. These modifications affect the efficiency and make the models more complex. In this thesis, a novel mixed integer programming based hyper-box enclosure approach is presented for multi-class data classification problems. In order to deal with large data sets, a three-stage mathematical programming based approach is developed for training part analysis of hyper-box enclosure method. Training set is preprocessed to identify the observations that are more difficult to classify, and seed finding and sub grouping algorithms are applied in the first stage. Then, optimization model is formulated considering these observations and seeds. Finally, assignments of non-problematic instances, intersection elimination and box combination algorithms are carried out. After training analysis with this three stage approach, the efficiency of the method is tested by the simple distance based testing algorithm. The efficiency of the proposed three-stage method is tested on two separate benchmark problems; the protein folding type prediction problem and the UCI Repository data sets. The computational results on the illustrative example and the benchmark problems show the accuracy of the proposed method.

ÖZET

Veri Sınıflandırma, farklı özelliklere sahip örneklerin bilinen sınıflara olan üyeliğini belirlemeye çalışan önemli bir veri madenciliği problemidir. Veri setini iki gruba ayıran geleneksel yöntemleri çok sınıflı veri sınıflandırma problemlerine uygulayabilmek için bazı düzenlemelere gerek vardır. Yapılan bu değişiklikler kullanılan yöntemin verimliliğini etkilemekte ve modeli daha karmaşık bir hale getirmektedir. Bu tezde çok gruplu veri sınıflandırma problemi için geliştirilmiş tamsayı karışık programlamaya dayalı yeni çok boyutlu kutu yaklaşımı anlatılmaktadır. Büyük veri kümeleri ile çalışabilmek için çok boyutlu kutu yaklaşımının eğitici bölümünde kullanılmak üzere üç aşamalı matematiksel programlamaya dayalı bir yöntem geliştirilmiştir. Birinci aşamada, eğitici kümedeki sınıflandırması zor olan örnekler belirlenerek, tohum bulma ve alt küme oluşturma algoritmaları uygulanmaktadır. Daha sonra edinilen bu gözlem ve tohumlar kullanılarak eniyileme modeli çözülmektedir. Son olarak da problemsiz örneklerin kutulara atanması, kesişme engelleme ve kutu birleştirme algoritmaları uygulanmaktadır. Bu üç aşamalı eğitici çalışmalar sonrasında, metodun verimliliği uzaklığa dayalı basit bir test algoritması ile ölçülmüştür. Bu üç aşamalı modelin verimliliği veri sınıflandırılmasında çok bilinen ve çok kullanılan veri setleri üzerinde test edilmiştir. Bunlar protein katlanma tahmin problemi ve UCI veri havuzu problemleridir. Örnek problem ve bilenen veri setleri kullanılarak elde edilen sonuçlar önerilen yöntemin çok sınıflı veri sınıflandırma problemine önemli bir katkıda bulunduğunu kanıtlamaktadır.

ACKNOWLEDGEMENTS

Pursuing a Ph.D. thesis is both painful and enjoyable experience. It is just like climbing a high peak, step by step, accompanied with bitterness, hardships, encouragement, trust and with so many people's kind help. When I found myself at the top enjoying the beautiful scenery, I realized that there are many people whose enormous contribution enabled the completion of this journey.

First of all, I would like to express my gratitude to my Ph.D. advisor Assoc. Prof. Metin Türkay for providing me the opportunity to work in such an exciting and inspiring project. He taught me how to write academic papers, made me a data mining person and had confidence in me. More importantly, he taught me how to work hard and play hard by myself.

Besides my advisor, I would like to thank the rest of my Ph.D. thesis committee: Assist. Prof. Deniz Yüret who asked me hard questions and gave insightful comments and guidance through my progress report presentations, Assos. Prof. Ceyda Oğuz for her supportive comments and participation in my follow up committee. I also thank Prof. Dr. Kuban Altınel and Prof. Dr. Serpil Sayın for participation in my PhD thesis committee, for their comments and future extensions on my Ph.D. thesis.

I also thank my lab members who made the Systems Lab a wonderful workplace for the past four years: Uğur Kaplan for his knowledge, interesting discussions, advices and brotherhood, Ali Öztürk for being my fellow sufferer and for his intelligent discussions, Pınar Kahraman-Öztürk for her close friendship and providing support. I am also grateful to all my friends, Bora, Pelin, Eda, Ethem, Selim, Can, Özge and Fatih for being funny & good classmates and officemates. I hope there will be other chances to work together again.

I want to give very warm wishes to my colleagues Semra and Caner who have followed my adventure from afar, in Gainesville, Florida: thanks for always being there and helpful advices. I especially thank to our valuable family friends Zeynep and Berkin for enjoyable dinners and brunches. I also thank my life long friends Selma and Feray for providing support and friendship that I needed.

Finally, I am very grateful to my large family: my parents, my brothers and their family, my mother-in-law, my father-in-law, my brother-in-law for their invaluable support and understanding. I am very lucky to be part of such a wonderful family. Last, but not least, I am greatly indebted to my devoted husband, Gökhan, for always believing in me and for his support at every step of my life. He was always there to listen and to give advice on any topic. I can not do anything without his patience, assistance, encouragement and love. I dedicate my Ph.D. thesis to my husband, Gökhan, and my family, to honor their love, patience, and support during these four years.

This thesis was completed with the support of the TUBITAK scholarship.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

CRM	Customer Relationship Management
NN	Neural Networks
SVM	Support Vector Machines
DAG	Direct Acyclic Graph
MP	Mathematical Programming
LP	Linear Programming
MILP	Mixed Integer Linear Programming
K-NN	K-Nearest Neighbor
RST	Rough Set Theory
LOO	Leave-one-out
10FCV	10 Fold Cross-validation
k	Class type
C_k	Number of correctly classified instances in class k
NC_k	Number of correctly classified instances not in class k
U_k	Number of under-predicted instances in class k
O_k	Number of over-predicted instances in class k
MCC	Mathews Correlation Coefficient
FFNN	Feed Forward Neural Network
RBF	Radial Basis Function
SOM	Self Organizing Maps
ART	Adaptive Resonance Theory
MSD	Minimization of the Sum of the Deviations
MMD	Maximization of the Minimum Deviation
D&C	Divide and Conquer

NMR	Nuclear Magnetic Resonance
PDB	Protein Data Bank
SCOP	Structural Classification Of Proteins
i	Training samples (*i=Sample1, Sample2, ..., SampleI*)
j	Test samples (*j=Sample1, Sample2, ..., SampleI*)
k	Class types (*k=Class1, Class2, ..., ClassK*)
l	Hyper-box that enclose a number of data points belonging to a class (*l=1,..,L*)
m	Attributes (*m=1,..,M*)
n	Bounds (*n=lo, up*)
ε	arbitrarily small positive number
Q	Large parameter
M	Total number of attributes
L	Total number of hyper-boxes
I	Total number of instances
N	Total number of bounds
K	Total number of classes
a_{im}	value of the attribute m for the sample *i*
D_{ik}	class *k* that the data point *i* belong to
yb_{l}	Binary variable to indicate whether the box *l* is used or not
ypb_{il}	Binary variable to indicate whether the data point *i* is in box *l* or not
ybc_{lk}	Binary variable to indicate whether box *l* represent class *k* or not
ypb_{ik}	Binary variable to indicate whether the data point *i* is assigned to class *k* or not
$ypbn_{ilmn}$	Binary variable to indicate whether the data point *i* is within the bound *n* with respect to attribute *m* of box *l* or not

$ypbm_{ilm}$	Binary variable to indicate whether the data point *i* is within the bounds of attribute *m* of box *l* or not
yp_{ik}	Boolean variable to indicate the misclassification of data point *i* to class *k*
X_{lmn}	the continuous variable that models bounds *n* for box *l* on attribute *m*
$XD_{l,k,m,n}$	the continuous variable that models bounds *n* for box *l* of class *k* on attribute *m*
DPI	Determination of Problematic Instances
D	Data Set
NI_k	Number of instances in class *k*
$DB_{ii'}$	Distance between two instances *i* and *i'*
S_i	Similarity of instance *i*
DS_i	Dissimilarity of instance *i*
SP_i	Binary variable to indicate whether the instance *i* is selected for this sub group or not
SS	Number of instances that exist in each of the constructed sub groups
TS	Number of subgroups
$PP_{ii'}$	Distance between instance *i* and *i'*
YP_i	Binary variable that indicates whether instance *i* is selected as seed or not
l'	Hyper-boxes that are obtained by combinations of the existing hyper-boxes
BC_{lk}	Class *k* of hyper-box *l* belongs to
$NX_{l'mn}$	Bounds *n* of hyper-box *l'* for attribute *m*
$NBC_{l'k}$	Class *k* of hyper-box *l'* belongs to
C_{lm}	Center of hyper-box *l* for attribute *m*
$C_{l'm}$	Center of hyper-box *l'* for attribute *m*
L_{lm}	Length of hyper-box *l* for attribute *m*
$L_{l'm}$	Length of hyper-box *l'* for attribute *m*

$IN1_{ll'm}$	Binary variable to indicate the intersection of hyper-box l with hyper-box l' for attribute m
$IN2_{ll'm}$	Binary variable to indicate the intersection of hyper-box l' with hyper-box l for attribute m
$IO_{ll'}$	Binary variable that represents the intersection of hyper-box l and hyper-box l'
$CO_{l'}$	Binary variable that represents an intersection related to hyper-box l'
$SO_{l'}$	Binary variable that indicates that hyper-box l' could be obtained without causing any intersection
$SI_{ll'}$	Binary parameter that gives the relationship with hyper-box l and hyper-box l'
$SN_{l'l''}$	Binary parameter that represents the relationship between hyper-box l' and hyper-box l''
DH_{il}	Minimum distance between instance i and the normal of hyper-box l
EP_{lj}	Extreme point j of hyper-box l
EP_{lt}	Extreme point t of hyper-box l
EPP	Set of extreme point combinations
ep_{ljm}	The value of attribute m for extreme point j of hyper-box l
ep_{ltm}	The value of attribute m for extreme point t of hyper-box l
w_{iljtm}	Difference between a_{im} and ep_{ljm}
v_{iljtm}	Difference vector between ep_{ljm} and ep_{ltm}
$C1_{iljtm}$	Dot product of w_{iljtm} and v_{iljtm}
$C2_{iljtm}$	Dot product of v_{iljtm} by itself
b_{iljtm}	Ratio of $C1_{iljtm}$ to $C2_{iljtm}$

pb_{iljtm}	Point where a_{im} is perpendicular to the edge between two extreme points
DED_{il}	Minimum distance between instance *i* and the edges of hyper-box *l*
DEP_{il}	Minimum distance between instance *i* and the extreme points of hyper-box *l*
$NDist_{il}$	Minimum distance from instance *i* to hyper-box *l*

Chapter 1

INTRODUCTION

Customer information becomes very important for companies as it is necessary to achieve power and success in the market. Due to recent advances in sophisticated hardware and software technologies, large quantities of data can be acquired, processed and stored. However, the amount of collected data frequently increases and constitutes large complicated databases. As a result of these structures, database management and data mining studies receive considerable attention. Data mining is the process of investigating and extracting implicit, previously unknown and potentially useful information form large data by using one or more computer-based learning techniques. The objective of data mining is to discover general patterns and similar characteristics of available data. Many different data mining methods exist; for example clustering, classification, association analysis, feature selection and characterization. Of these methods, data classification is the most important and widely studied topic [1].

1.1 Data Classification

Data classification, sometimes referred as pattern recognition or discriminant analysis, is a supervised learning strategy that analyzes the organization and categorization of data in distinct classes [2]. Generally, a training set, in which all objects are already associated with known class labels, is used by classification methods. The data classification algorithm works on this set by using the input attributes and builds a model to classify new objects. In other words, the algorithm predicts output attribute values. Output attribute of the developed model is categorical. For instance, a bank could attempt to understand the behavior of its customers via credit analysis, and customers can be assigned one of three possible labels; “safe”, “risky”, and “very risky”. The generated model could be used either to accept or reject future credit requests [1].

Classification has several significant differences from clustering, a related data mining technique. The class labels and the number of classes are not known in clustering. On the other hand, the class labels and the number of classes are known *a priori* for classification. In addition, there is no output attribute in clustering, thus, clustering algorithms attempt to group instances into two or more classes by using some measure of cluster quality [3]. Unlike clustering, prediction has an output attribute. However, the purpose of prediction is to determine future outcome rather than current behavior. In classification, an output attribute is categorical, whereas the output attribute of a predictive model can be either categorical or numerical. In summary, classification places emphasize on building models that are able to assign new instances to one of a set of well-defined classes [2].

There are many applications of data classification in finance [2, 3], health care [2], sports [2], engineering [2, 4], and science [4]. In finance, especially in risk management, data classification is applied to determine insurance rates, manage investment portfolios, and differentiate between individuals who have good or poor credit risks [3]. Furthermore, financial institutions use data classification to detect which customers are using which products so they can offer the right mix of products and services to better meet customer needs. Another application used by financial institutions is fraud detection in credit card and large cash transactions [2].

Additionally, several health care studies such as medical diagnoses and treatment effectiveness can be analyzed by the help of classification [2]. For instance, information about patients who have had or not yet had a heart attack is collected. A person’s risk for heart attack can be predicted using data classification methods. By considering these risk values, precautions are taken and certain medical treatments are applied to high risk patients [2].

In case of sports, data classification studies are carried out for horse racing and lottery. Data related to past matches between the teams are collected. Then, while playing chance games, gamblers use these past data and estimate the result of the future match and the winner [2].

Customer Relationship Management (CRM) is a well-known application of data classification in business that involves the management of interactions with customers [3]. For this purpose, information related to each customer is collected and this data are used to increase the efficiency of interaction with the customers in all stages. In CRM, classification is generally used to assign a score to a particular customer or prospect indicating the likelihood that the individual will behave in such a way that revenues and customer satisfaction levels are improved. For example, the inclination to respond to a particular offer or to switch to a product from a competitor could be measured by a score. Moreover, characterization of customer segmentation into groups with similar behavior, such as buying a particular product, can be identified by classification. Consequently, data classification models can add tremendous value to organizations both in finance and business [2, 3].

Data classification has a wide range of security related applications as well: fingerprint and facial recognition are the most studied topics. Another widely used application of data classification is in the area of bioinformatics; classification methods are being used in order to get valuable information on the characteristics of genes and proteins. Many classification methods are used in micro array analysis to predict sample phenotypes based on gene expression patterns [4]. Another problem in bioinformatics that attracted a lot of attention in the literature is the prediction of secondary structure of a protein from its amino acid sequence [4]. Moreover, protein folding type prediction is also studied with different classification methods [4]. In conclusion, data classification is an important problem that has applications in a diverse set of areas ranging from finance to bioinformatics.

1.2 Data Classification Methods

Typical classification algorithms have three basic steps; model construction, model evaluation, and model use [1]. Each instance in training set is assumed to belong to a predefined class. By examining input attributes of the training samples, a classification model defining the general characteristics of existing classes is obtained during the model construction step. Depending on the solution approach, the model can be represented in different forms such as mathematical formulae, rule, or a computer program. The next step, model evaluation, is the accuracy estimation of the model based on a test set. In this evaluation part, known labels of each of the test samples are compared with the results of the model. The percentage of test set samples that are correctly classified by the model constitutes the accuracy value of the method. Selecting the instances of the test set is very critical: the test set must be independent of training set in order to obtain reliable results. Finally, if the accuracy of the developed model is preferable, then it is used to classify the unseen samples by assigning labels for them.

A broad range of methods exists for data classification problems including Neural Networks (NN), Support Vector Machines (SVM), Mathematical Programming, Decision Trees, *K*-nearest Neighbor, Logistic Regression, Bayesian Networks, Genetic algorithms, Rough Set Theory, and Fuzzy Sets. An overall view of classification methods is published by Weiss and Kulikowski [5]. In this study, available classification and prediction methods from statistics, neural networks, machine learning and expert systems are reviewed.

Widely studied data classification methods are explained briefly in the following subtitles.

1.2.1 Neural Networks

A neural network is a data structure that attempts to simulate the behavior of neurons in a biological brain. While the human brain consists of billions of neurons, a typical neural network is composed of layers of interconnected nodes up to 100. From one unit to another, messages are passed along these connections. Through this transfer, a message can change based on the weight of the connection and the value in the node. Neural networks operate in two phases; learning and output. During the network learning, attribute values of the training instances enter the network at the input layer. The network connection weights and attribute values are practiced to compute the output for each training instance. These output values are compared with the desired network output and any error between these two values is calculated to modify the weights of the interconnections. Learning phase terminates after a predetermined number of iterations or minimum error rate is achieved. Finally, network weights are fixed and the network is used to compute output values for new instances in the output phase [2].

A major shortcoming of the neural network approach is a lack of explanation of established model. Moreover, converting categorical values to numerical ones could be a challenging issue. In addition, although the prediction accuracy is generally high, neural networks need long training times [4, 6]. Moreover, the training procedures can lead to both over fitting problem [7, 8] and gets stuck at a local optimum of the cost function.

1.2.2 Support Vector Machines

Support Vector Machines (SVM) is a new classification technique developed by Vapnik and his group [9]. They operate by finding a hyper surface that will split the classes so that the distance between the hyper surface and the nearest of the points in the groups has the largest value. The main goal is to generate a separating hyper surface which maximizes the margin and produces good generalization ability [4]. In recent years, SVM has been considered one of the most efficient methods for two-class classification problems [10].

On the other hand, the SVM has some important drawbacks. First, a combination of SVMs has to be used in order to solve the multi-group classification problems. Second, some approximation algorithms are used in order to reduce the computational time for SVMs while learning the large scale of data. However, this computational improvement could cause less efficient performance values. Additionally, choice of the Kernel Function and the values of parameters are important decisions that directly affect the performance.

To overcome the above problems, many variants of SVM have been suggested including the use of SVM ensemble with bagging or boosting rather than the use of a single SVM [11]. Hsu *et al.* [12] compared the performance values of “all-together” and binary classification based methods such as “one-against-all”, “one-against-one” and direct acyclic graph (DAG) SVM.

The one-against-all method is the earliest used implementation for SVM multi-class classification. It constructs k SVM models where k is the number of classes. The i^{th} SVM is trained with all of the examples in the i^{th} class with positive labels, and all other examples with negative labels. One piece at a time each class is separated from the others.

Conversely, one-against-one method constructs $k(k-1)/2$ classifiers where each one is trained on data from two classes. In the testing part, if sign of the model says x is in the i^{th} class, then the vote for the i^{th} class is added by one. Otherwise, the j^{th} is increased by one. Finally, x is predicted to be in the class with the largest vote.

Direct acyclic graph SVM method’s training phase is the same as the one-against-one method by solving $k(k-1)/2$ binary SVMs. However, in the testing phase, it uses a rooted binary directed acyclic graph which has $k(k-1)/2$ internal nodes and k leaves. Each node is a binary SVM of i^{th} and j^{th} classes. Given a test sample x, starting at root node, the binary decision function is evaluated. It then moves either left or right depending on the output value. Therefore, it goes through a path before reaching a leaf node which indicates the predicted class.

Hsu *et al.* [12] conclude that "one-against-one" and DAG binary classification methods are more suitable for practical use than the other methods. Nevertheless, for solving multi-class SVM in one step, a much larger optimization problem is required so experiments are limited to small data sets.

1.2.3 Mathematical Programming Approaches

The mathematical programming approach to linear discriminant analysis was first introduced in early 1980's. Since then, numerous mathematical programming models have appeared in literature. As an extension of complement to these, Erenguc and Koehler made a comprehensive review [13]. In their research, they formulate a typical mathematical programming (MP) approach as follows:

$$\text{minimize} \quad f(w,c) \tag{1.1}$$

$$\text{subject to:} \quad X_1 w \leq c1 \tag{1.2}$$

$$X_2 w \geq (c + \varepsilon)1 \tag{1.3}$$

$$w \neq 0 \tag{1.4}$$

By this general formulation MP approach tries to determine a scalar c and a non-zero vector $w \in R^p$ such that the hyper plane $w'x = c$ partitions the m-dimensional (m: the number of attributes) Euclidean space R^m into a closed half-space $w'x \leq c$ and an open half-space $w'x > c$. In the formulation, ε represents an arbitrarily small positive number. An interior and exterior deviation term for each group are defined for MP approaches. An interior deviation is the deviation from the hyper plane of a properly classified point. An exterior deviation is the deviation from the hyper plane of an improperly classified point.

Many distinct MP methods with different objective functions are developed in literature. These include; minimizing the maximum exterior deviation, minimizing the weighted sum of exterior deviations, minimizing a measure of exterior deviations while maximizing a measure of interior deviations, minimizing the number of misclassifications, and minimizing a generalized distance measure. Most of these methods modeled data classification as linear programming (LP) problems which optimize a distance function. Contrary to LP problems, mixed-integer linear programming (MILP) problems with minimizing the misclassifications on the design data set are also widely studied [12].

MP methods have certain advantages over the parametric ones. For instance, they are free from parametric assumptions and weights to be adjusted. Moreover, varied objectives and more complex problem formulations can easily be accommodated by using MP methods. On the other hand, obtaining a solution without any discriminating power, unbounded solutions and excessive computational effort requirement are some of the problems in MP based methods.

1.2.4 Decision Trees

Decision Trees are one of the most popular top-down induction techniques in data classification. One of the main reasons behind this popularity appears to be their transparency and relative advantage in terms of interpretability. Moreover, there exist two powerful implementations of decision trees; CART [14] and C4.5 [15]. Most decision tree induction algorithms construct a tree in a top-down manner by selecting attributes one at a time and splitting the data according to the values of those attributes. The most important attribute is selected as the top split node, and so forth. For example, in C4.5 attributes are chosen to maximize the information gain ratio in the split [15]. The basic steps of a decision tree algorithm are as follows [2]:

1. Let *T* be the set of training instances.
2. Choose an attribute that best differentiates the instances contained in *T*.
3. Create a tree node whose value is the chosen attribute. Create child links from this node where each link represents a unique value for the chosen

attribute. Use the child link values to further subdivide the instances into subclasses.

4. For each subclass created in step 3:
    a. If the instances in the subclass satisfy predefined criteria or if the set of remaining attribute choices for this path of the tree is null, specify the classification for new instances following the decision path.
    b. If the subclass does not satisfy the predefined criteria and there is at least one attribute to further subdivide the path of the tree, let *T* be the current set of subset classes and return to step 2.

Existing decision tree algorithms are computationally efficient and practically successful. However, the fact that they are limited to constructing axis-parallel separating planes limits their effectiveness in applications where some combinations of attributes are highly predictive of the class [16]. A further drawback lies in the fact that continuous variables are implicitly discretized by the splitting process, losing information along the way. Moreover, most decision tree algorithms are known to be unstable when dealing with a large data set where it can be impractical to access all data at once and construct a single decision tree [17].

1.2.5 *K*-Nearest Neighbor Algorithm

The nearest neighbor method is a non-parametric classification technique proposed by Fix and Hodges [18] and then modified by Cover and Hart [19]. The *K*-nearest neighbor (*K*-NN) classifies unlabeled samples based on their similarity with the observations in the training set. Thus, for a given unlabeled sample, we find the "*K*-closest" labeled observations in the training set and assign the unlabeled samples to class that appears most frequently within *k* subset. Experimental studies show that *K*-nearest neighbor is computationally expensive for a large data set, but it is simple and running faster than other classification methods. Moreover, the misclassification rate of *K*-NN rule approaches the optimal error rate asymptotically as *k* increases.

The *K*-NN algorithm uses the metric properties of the data space. The most commonly used metrics in measuring the distance of a sample from a given training set $X \equiv [x_1, \quad x_2, \quad \cdots \quad x_m]$ are as follows:

- Euclidean Distance:

$$d_2(X, X^*) = \sqrt{\sum_{i=1}^{m} (x_i - x_i^*)^2} \tag{1.5}$$

- Minkowski Distance:

$$d_q(X, X^*) = \sqrt[q]{\sum_{i=1}^{m} \left| x_i - x_i^* \right|^q} \tag{1.6}$$

- Elliptical Distance:

$$d(X, X^*) = \sum_{i=1}^{m} \left| x_i - x_i^* \right| 2 \tag{1.7}$$

The major weakness of *K*-nearest neighbors lays in both choices the value of *k* and calculation of case neighborhood: for this one, one needs to define a metric that measures the distance between data items. In most application areas, it is not clear how to, other than by trial and error, define a metric in such a way that the relative importance of data components is reflected in the metric. Furthermore, as the size of the training set becomes large, distance calculation process becomes very expensive. Moreover, it needs a large storage, because it runs using the entire training set and highly sensitive to the curse of dimensionality.

1.2.6 Logistic Regression

Logistic Regression is a nonlinear regression technique that associates a conditional probability score with each data instance [2]. It is useful when the dependent variable is either binomial or multinomial values. Binomial logistic regression is a form of regression which is used when the dependent variable is a binary and the independent variables are continuous, categorical or both. On the other hand, multinomial logistic regression exists to handle the case of more than two dependent variables [20].

Generally, logistic regression produces a formula that predicts the probability of the dependent variable as a function of the independent variables. It produces Odds Ratios (Equation 1.8) by the help of the term *p(k=1|x)*, the probability of seeing the class associated with $k = 1$ given the values contained in the feature vector x. As it is producing *odds ratios* as functions of predictors, the regression coefficient in the logistic regression model has no interpretation of the linear correlation.

$$\frac{p(k=1 \mid x)}{1 - p(k=1 \mid x)} \tag{1.8}$$

For any feature vector x, the odds indicate how often the class associated with $k = 1$ is seen relative to the frequency in which the class associated with $k = 0$ is observed for the binomial case. After taking the natural log of this odds ratio and some transformations, logistic regression model given in Equation 1.9 will be obtained. The method iteratively tries to determine the coefficient values for the exponent term $ax+c$ in Equation 1.9. Convergence occurs when the logarithmic summation is close to zero or when the value does not change from one iteration to the next [2].

$$p(k=1 \mid x) = \frac{e^{ax+c}}{1 + e^{ax+c}} \tag{1.9}$$

1.2.7 Bayesian Networks

Bayes classifier is a simple but powerful data classification technique. The model assumes all input attributes to be of equal importance and independent of one another. The classifier is based on Bayes Theorem given in Equation 1.10 where *H* is a hypothesis to be tested and *E* is evidence associated with hypothesis. Hypothesis is the dependent variable and represents the class. The evidence is determined by input attributes. *P(E|H)* is the conditional probability that *H* is true given evidence *E*. *P(H)* is an a priori probability, which denotes the probability of the hypothesis before any evidence is given [2].

$$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)} \tag{1.10}$$

A Bayesian network is a directed acyclic graph *G* that model probabilistic relationships among a set of random variables where each variable has specific classes. Each node in the graph represents a random variable and each edge captures the direct dependencies between variables. The network encodes the conditional independence relationships that each node is independent of its non-descendants given its parents [21]. The popular Bayesian network implementation is Naïve Bayes method.

1.2.8 Other Methods

Genetic Algorithms are used in data classification problems that are difficult to solve using conventional methods. It is based on Darwinian principle of natural selection; crossover and mutation are the most widely used genetic operators. In a basic genetic learning algorithm, a population *P* of *n* elements is initialized which often referred to as chromosomes. A fitness function is used to evaluate each element of current solution. If an element passes fitness criteria, it remains in *P*. By using genetic operators new elements are created and added to the population. This procedure is carried on until a specified termination condition is satisfied [4].

Rough Set Theory (RST) can be approached as an extension of the Classical Set Theory [2]. Rough sets are considered as the sets with fuzzy boundaries, in other words the sets that cannot be precisely characterized using the available set of attributes. In data classification, it is inconvenient to describe the similarity among data with the indiscerniblity relation because two data *x* and *z* cannot be guaranteed in the same class even though a couple of data *x* and *y* are contained in the same class and another couple of data *y* and *z* are also contained in the same class. In other words, the transitivity property is not always useful in the problem of data classification. This non-transitivity property is more salient for the data within the boundary region. For this reason, a tolerant relation appropriate for the data classification problem is studied by some researchers.

In contrast, Fuzzy Sets are based on Fuzzy Logic [4]. Fuzzy logic is an extension of Boolean logic (YES or NO) dealing with the concept of partial truth. Whereas, classical logic holds that everything can be expressed in binary terms (0 or 1, yes or no), fuzzy logic replaces Boolean truth values with degrees of truth.

1.3 Performance Evaluation

In evaluating the performances of classification methods, the percentage of correctly classified instances, accuracies, are estimated and compared. Accuracies estimated on the training set are called as self-consistency results. It is widely known that self-consistency test results tend to be biased. Hence, two different error estimation methods are recommended to have unbiased performance evaluation.

1.3.1 Training and Test Sets

Training set is a sample of data that is used to build classification rules and functions. In order to test the performance of the classification method, another independent data set, test set, is used. True classes of the instances in that test set are known but are not shown to the classifier. Finally, predicted and true classes of test set instances are compared and classification performance is estimated by the number of correctly classified instances. As test set instances are unseen by the classifier, this performance estimate is unbiased. When a data set is given, conventionally a 2/3 of the data set is selected randomly and used as test set. The classifier is trained on the remaining data and then tested on the test data. There exists a small loss of efficiency due to not use the full sample as training but this is not a major problem for large data sets. Generally, this type of performance evaluation is adopted when the number of instances are much larger than 1000 [22].

1.3.2 Cross-validation

For moderate size samples, the cross validation is preferred. In cross-validation, data set is divided into *m* equal-sized sub samples randomly. Each sub sample is treated as a test set and predicted via the classification rule constructed from the remaining (*m*-1) sub samples. The classification performance is estimated by taking the average of these *m* sub samples. In this way, the classification rate is calculated efficiently and in an unbiased way. Leave-one-out (LOO) rate is simply applying the cross-validation with *m* equal to the number of instances. LOO and 10 fold cross-validation (10FCV) are very popular performance evaluation methods [22].

1.3.3 Sensitivity and Specificity

In classification methods, giving only the accuracy values are not sufficient to analyze the results. There exist other values to be estimated and analyzed such as sensitivity, specificity, Mathews Correlation Coefficient and performance with respect to random prediction. In order to define these values easily, a representative confusion matrix given in Table 1.1 will be used. The values *a*, *b*, *c* and *d* are the number of correct predictions for the respective classes 1, 2, 3, and 4. Moreover, *ab* is the number of incorrect predictions where Class 1 instance is predicted as Class 2 and *ba* is the number of incorrect predictions where Class 2 instance is predicted as Class 1. The other values of the confusion matrix are similar definitions with *ab* and *ba*.

Table 1.1 A representative confusion matrix for a four-grouped data classification problem.

ACTUAL	PREDICTED CLASSES			
CLASSES	Class 1	Class 2	Class 3	Class 4
Class 1	*a*	*ab*	*ac*	*ad*
Class 2	*ba*	*b*	*bc*	*bd*
Class 3	*ca*	*cb*	*c*	*cd*
Class 4	*da*	*db*	*dc*	*d*

Furthermore, in order to simplify the equations of performance measures, we need to define five more parameters. Total number of instances in the data set is symbolized by *N*. In Table 1.1, *N* will be total sum of the values in each of the rows and columns of the confusion matrix. C_k represents the correctly classified instances in class *k*. For example, in Table 1.1, C_1 will equal to *a*. NC_k is used to give the number of correctly classified instance not in class *k*. In Table 1.1, NC_2 will equal to $(a+c+d)$. Additionally, the number of under-predicted instances and over-predicted instances for class *k* are defined by U_k and O_k, respectively. U_3 will be the sum $(ac+bc+dc)$ and O_3 will be the sum $(ca+cb+cd)$ from Table 1.1. Using these four new parameters, other performance measure definitions will be much simpler.

The sensitivity is the ratio of correct and all predictions for a given structural class [23]. The sensitivity value of class k is given in Equation (1.11).

$$Sensivity_k = \frac{C_k}{C_k + O_k} \tag{1.11}$$

The specificity is the ratio between the correct and all predictions for proteins that should be excluded for a given class [23]. The specificity value of class *k* is given in Equations (1.12).

$$Specificity_k = \frac{N - C_k - O_k - U_k}{N - C_k - O_k} \tag{1.12}$$

Generally, average specificity and sensitivity values are given for classification methods. These values can be calculated taking the weighted averages of individual specificity and sensitivity values with respect to the class sizes. In Equations (1.13) and (1.14), formal definitions of average sensitivity and specificity values are presented, respectively.

$$Sensivity = \sum_k \frac{C_k + O_k}{N} sensivity_k \tag{1.13}$$

$$Specificity = \sum_k \frac{C_k + O_k}{N} specificity_k \tag{1.14}$$

1.3.4 Mathews Correlation Coefficient

Mathews Correlation Coefficient (MCC) is a limited number between -1 and 1. If there is no relationship between the predicted values and actual values, the MCC should be 0 or very low (the predicted numbers are not better than random numbers). In contrast, the MCC value would increase as the strength of the relationship between the predicted values and actual values increases. It is obvious that a perfect fit gives a coefficient of 1. The higher MCC indicates the better performance of the prediction [24]. The MMC value for class *k* can be calculated using Equation (1.15).

$$MCC_k = \frac{\left[C_k NC_k - U_k O_k\right]}{\sqrt{(C_k + U_k)(C_k + O_k)(NC_k + U_k)(NC_k + O_k)}} \tag{1.15}$$

1.3.5 Performance with Respect to Random Prediction

Performance with respect to random prediction can be calculated by Equation (1.17). For a perfect prediction, S_k should be equal to 1 while for the predictions that are no better than random it would be equal to zero [24].

$$RTotal_k = \frac{(C_k + U_k)(C_k + O_k) + (NC_k + U_k)(NC_k + O_k)}{N} \tag{1.16}$$

$$S_k = \frac{C_k + NC_k - RTotal_k}{N - RTotal_k} \tag{1.17}$$

Besides giving the accuracy values of the studied data sets, we will investigate these performance measures and analyze the results deeper.

1.3.6 P-value Analysis

When comparing supervised classification models, the *P*-value (paired *t*-test) analysis based on hypothesis testing need to be carried out in order to examine the differences in a statistical manner. *P*-value represents the difference between two models with 95% confidence. If *P*-value is greater than 2, the difference between the results of the models is not due to chance. Otherwise, the accuracies of the models are very close to each other and no significant improvement achieved. *P*-value can be calculated using Equation (1.18). In this equation, E_1 and E_2 are the error rates of two models; q is the average of two error rates; n_1 and n_2 are the number of instances in the test sets of two models.

$$P = \frac{|E_1 - E_2|}{\sqrt{q(1-q)(1/n_1 + 1/n_2)}} \tag{1.18}$$

1.4 Ideal Characteristics of Classification Methods

While evaluating the data classification methods, some important properties of the model have to be considered in detail. Firstly, methods are usually evaluated on the test data. Prediction accuracy, ability of the model to correctly predict the class label, is a very considerable point for evaluation. Most of the comparisons between the models are done by looking directly to these prediction accuracy values. On the other hand, time to construct the model and time to use it also has a big role in real life applications. For a preferable data classification model, computational time must be reasonable. Thirdly, for an ideal data classifier, it should have a few parameters to tune in the system as possible. In Neural Networks, the weights between the nodes have to be adjusted. Since all of the existing weights need to be optimized, it is not easy to incorporate the domain knowledge and they possess a long training time. Moreover, it is difficult to understand the learned function. Similarly, SVM method has the biggest limitation of choosing the kernel function. Once the kernel is fixed, SVM classifiers have only one user-chosen parameter, error penalty. However, kernel is a very important decision criterion. Another important characteristic of an ideal data classifier is the ability to form a decision boundary that minimizes the amount of misclassification for all of the overlapping classes in the training set.

Some of the methods mentioned above can only be used for the two class cases, such as yes (class1) or no (class 2). However, the number of classes to be classified is generally more than two in real life problems. Existing methods can be somehow modified or developed for multi-class case. In that situation, the accuracy values of the models decrease [4]. For instance, SVMs are originally a model for two class problems and are more effective. For multi-class case, combinations of SVMs should be used. Since SVMs use some approximation algorithms in order to reduce the computational time, increasing number of these approximation algorithms causes the degradation of classification performance. Thus, the performance does not improve as much as in binary case. Therefore, there is a need for new approaches that are able to address multi-group problems effectively. In this study, a novel mixed-integer programming approach for multi-class data classification problem has been developed. The proposed approach is based on the use of hyper-boxes for defining boundaries of the classes that include all or some of the points in that set. The computational results on the studied datasets show that the suggested method is accurate and efficient on multi-class data classification problems.

1.5 Contributions

This thesis presents a novel three-stage mathematical programming based hyper-box enclosure approach for multi-class data classification problems. A mixed-integer programming model is developed for representing existence of hyper-boxes which define the boundaries of the classes for the training set. In order to overcome the computational difficulties for large data sets, a three-stage approach is developed for training part analysis of hyper-box enclosure approach. The performance of the model is tested by applying the testing part of the proposed method. Main contributions of this thesis can be summarized as follows:

One of the most important contributions is that the proposed data classification method based on mixed-integer programming allows the use of hyper-boxes for defining boundaries of the classes that enclose all or some of the points in that set. This approach in the training problem can indirectly effect and improve the prediction accuracy of the model. This may be one of the reasons behind the high classification accuracy values obtained by the proposed model.

The suggested model can be used for both binary and multi-class cases without any modifications or additions. High classification accuracies are observed for binary and multi-class problems.

The proposed model has only one parameter to initialize (big-M parameter) and this parameter does not require adjusting during the training of the model. Furthermore, the model can operate without a priori knowledge about the underlying distribution of the data.

From the computational time perspective, the proposed three-stage MILP approach is applicable to obtain solutions to large multi-class data classification problems.

Furthermore, the testing algorithm is computationally tractable for high dimensional data sets. As observed from the examined data sets, total computational time for proposed approach is reasonable and less than the other methods used for these data sets.

The proposed approach in this thesis gives high accuracy values on the studied benchmark data sets. Hence, the developed multi-class data classification model is at least as accurate as the other models including NN, SVM, Decision Trees, *K*-Nearest Neighbor, Logistic Regression, Bayesian Classifier, etc.

In summary, by the development of this new approach, solutions to multi-class data classification problems can be obtained and the prediction accuracies can be improved. In addition to this, the simplicity and the understandability of the proposed model are preferable.

1.6 Outline

This thesis contains six chapters. Chapter 2 provides a literature review on data classification summarizing distinct methods reported. Moreover, existing mathematical programming based approaches to data classification are investigated in detail. The literature on protein folding type problem is also mentioned in Chapter 2. The developed three-stage MILP based hyper-box enclosure approach to multi-class data classification is presented in Chapter 3. The mixed-integer programming formulation, sub grouping algorithm, seed finding algorithm, intersection elimination algorithm and box combination algorithm for the training part of the problem are discussed in detail. In addition, original and new testing algorithms are explained and compared. The method is also illustrated on a small illustrative example in Chapter 3. The application of the proposed approach on existing protein folding type benchmark data sets are illustrated and results are examined in Chapter 4. Furthermore, the efficiency of the proposed method on existing eleven UCI Repository benchmark data sets is tested and results are given in Chapter 5. The thesis is concluded with short summary, conclusions, and directions on future research work with Chapter 6.

Chapter 2

LITERATURE REVIEW

Data classification is a multidisciplinary problem which is a very active area of study and research. Classification problems have been intensively studied by a diverse group of researchers including statisticians, engineers, biologists, computer scientists. There are variety of methods for solving classification problems such as Neural Networks (NN), Support Vector Machines (SVM), Decision Trees, Bayesian Networks, Logistic Regression, *K*-nearest neighbor, tolerant rough sets, fuzzy logic and Mathematical Programming [25]. In this chapter, a literature review on data classification methods, mathematical programming based methods and an important problem, prediction of folding type of proteins, is provided.

2.1 Literature Review on Data Classification Methods

An overall view of classification methods is published by Weiss & Kulikowski [5]. In this book, available classification and prediction methods from statistics, neural networks, machine learning and expert systems are reviewed. Hand [26] investigates the statistical approach of data classification and pattern detection in the fields of medicine, psychology and finance. More recently, Webb provides an introduction to statistical pattern recognition theory and techniques in his book [27]. In that book, descriptions of today’s pattern recognition techniques including many of the recent advances in nonparametric approaches to data classification in the statistics literature are provided. Moreover, the techniques are illustrated with examples of real-world applications. The estimation of error rates in discriminant analysis is explored by Lachenbruch & Mickey [28]. In this study, leave-one-out cross-validations tests are proposed for error estimation. *N* (number of data points) separate times, the classification function is trained on all the data except for one point and a prediction is made for that point in leave-one-out cross-validation tests. Average error is computed and used to evaluate the model. The evaluation given by this cross-validation test error is good, but computing the result of leave-one-out tests takes very long time. Kendall *et al.* [29] give a comprehensive exposition about the statistical approach of data classification and advance theory of statistics. Furthermore, McLachlan studied on a thorough treatment of statistical procedures in discriminant analysis and pattern recognition [30].

The study by Hertz *et al.* [31] is one of the most detailed and reliable information guides for neural network approach in data classification. They propose an introduction to neural computation and explain the theory of the neural network approach. Additionally, Simpson [32] developed a fuzzy min-max classification neural network in which pattern classes are utilized as fuzzy sets. In this study, learning in the neural network was performed by properly placing and adjusting hyper boxes in the pattern space. Simpson defines a fuzzy set hyper-box as an *n*-dimensional box defined by a min and a max point with a corresponding membership function. The min-max membership function defines a fuzzy set, hyper-box fuzzy sets are aggregated to form a single fuzzy set class, and the resulting structure fits naturally into a neural network framework. Therefore, this classification system is referred as fuzzy min-max classification neural network. Since it uses only a min and a max point in the *n*-dimensional space and combines fuzzy sets with the neural network idea, this model has a different approach as compared to the proposed model in this thesis. Moreover, Zhang [33] gave a review of the use of feed-forward neural networks for classification. In data classification problems, neural networks have the ability to learn nonlinear input or output relationships while propagating and adopting itself with a given training set by training procedures. The learning process involves updating network architecture and connection weights in order to achieve efficiency by the help of some learning algorithms. The most common types of neural networks that are used for data classification are feed forward neural networks (FFNN) which includes multilayer Perceptron and Radial Basis Functions (RBFs) [34-37]. In FFNNs, the neurons are organized in different layers and each of the neurons in one layer can receive an input from units in the previous layers without loss of generality. On the other hand, RBF network is capable to perform a nonlinear mapping between the input and output vector space. It is widely used in data classification problems such as speech recognition, medical diagnosis, handwriting recognition, image processing, and fault diagnosis. The other popular network is Kohonen network (self organizing map (SOM)) [38] in which two dimensional discretized representation of the input space of the training samples are produced during the training phase. SOMs are different than other neural networks in the sense that they use a neighborhood function to preserve the topological properties of the input space.

On the other hand, Devijver & Kittler [39] concentrate on the *K*-nearest neighbor approach for data classification problems from the perspective of statistical approach. A comprehensive review of *K*-NN and many of the important contributions to the literature are included in Dasarathy [40]. The performance of the *K*-NN depends on the choice of k. If the value of k is larger, the procedure is more robust but needs more computation. Hans [41] mentioned that k must be smaller than the minimum of n_j, the number of observations in class j. Otherwise, the neighborhood is no longer the local neighborhood of the sample. Other choices of k are $n^{2/8}$, $n^{3/8}$, and $n^{1/2}$, subject to rounding up to the nearest integer, where n is the total number of observations in the training set [37]. While the optimal value of k depends on the size and nature of the data, typical values are 3, 5, or 7.

One of the first papers published on data classification introduces fuzzy adaptive resonance theory (ART) which is a fast and reliable analog pattern clustering system. In this study, Carpenter and Grossberg combine the fuzzy logic with the idea of ART and try to develop an efficient classifier [42]. A general neural-network model for fuzzy logic control and decision systems including the data classification problem is discussed in [14].

Rough set theory introduced by Pawlak [43] is a mathematical tool to deal with vagueness and uncertainty in machine learning and pattern recognition. Two applications of logic for classification using rough set approach are presented in [44]. The multi-model logics is employed for automatic feature selection while a rough-set-based inductive reasoning is used for discovering optimal feature set with respect to the quality of classification as well as for improving the performance of decision algorithms. Another approach in data classification is to use rough sets by tolerating the relationships among the objects for pattern classification [45]. A data classification method based on the tolerant rough set that combines the use of logic and the tolerance relation among the objects is presented in [46]. The performance of this approach is tested on the UCI Repository data sets [47]. Furthermore, Castro *et al.* [48] presented a method to learn maximal structure rules in fuzzy logic to deal with the one of the UCI Repository data sets, Iris. Chen *et al.* [49], Hong *et al.* [50], Lin *et al.* [51] and Wu *et al.* [52] presented different methods to generate fuzzy rules from training instances based on genetic algorithms to study UCI Repository data sets. Most recently, Chen *et al.* [53] developed a new model based on distributions of training instances. Their proposed method achieves a higher average classification accuracy rate than existing methods. On the other hand, Uney and Turkay [54] proposed a mixed-integer linear programming approach and tested the performance of the method on Iris data set.

The training procedure of support vector machines (SVMs) usually requires huge memory space and significant computation time due to the enormous amounts of training data and quadratic programming problem [55]. Some of the researcher proposed incremental training or active learning to shorten the training time [56]. The main idea is to select a subset of training samples while preserving the performance as using all the training samples. Syed *et al.* [57] and Campbell *et al.* [58] proposed two different incremental learning procedures. On the other hand, multi-group data classification problems are solved either by constructing several two class classifier such as one-against-one, one-against-all, and DAG SVMs [12] or by constructing multi-class classifier directly such as *k*-SVM [59]. Recently, Zhu *et al.* [60] proposed a multi-class classification algorithm which adopted the minimum enclosing spheres to classify a new example and showed that the resulting classifier performed comparable to the standard SVMs. Based on Zhu *et al.* [60], Wang *et al.* [61] and Lee *et al.* [62] also proposed a new classification rule on the basis of Bayesian optimal decision theory.

Mathematical optimization techniques have been applied directly in the optimal construction of decision boundaries in the decision tree induction. Bennett [63] introduced an extension of linear programming techniques to decision tree construction for two class problems. Kennedy *et al.* [64] first developed a genetic algorithm for optimizing decision trees. In their approach, a binary tree is represented by a number of unit sub trees each having a root node and two branches. When using genetic algorithm to optimize the tree, the growth of the tree could not be controlled as genetic algorithm does not evaluate the size of the tree. Therefore, the resulting tree may become overly deep and complex or may be too simple. To address this problem, Niimi and Tazaki [65] combine genetic programming with association rule algorithm for decision tree construction. In this approach, rules generated by apriori association rule discovery algorithm are taken as the initial individual decision trees for a subsequent genetic programming algorithm.

In summary, a large number of data classification methods have been developed up to now; however each of them has some drawbacks which make them unattractive. Thus, researchers have been studying to develop more accurate and more efficient methods or to improve the existing methods.

2.2 Literature Review on Mathematical Programming Based Methods

Mathematical Programming (MP) based data classification models are used to generate linear discriminant functions, or separating hyper-planes, which optimally separate observations in a training set. Generally, two group data classification problems are considered by MP techniques and they can be extended to multi-group problems [66, 67]. Erenguc and Koehler [13] summarized the existing mathematical programming models and their experimental results.

Mangasarian [68] is the first researcher who proposed a linear programming model to determine separating hyper-planes, namely linear discriminant function, for two linearly separable classes. In the case of linearly inseparable classes, Freed and Glover [67] proposed a mathematical model which tries to minimize the sum of the deviations (MSD) of misclassified instances from the separating hyper-plane. In addition to that, Hand [69] developed a mathematical model with an objective function of maximization of the minimum deviation (MMD) of the misclassified instances from the separating hyper-plane. For multi-group problems, a model based on goal programming was also suggested by Freed and Glover [70]. An alternative LP approach for multi-group data classification problems has been proposed in [71]. In addition to being non-parametric, LP and other MP based approaches are also more flexible than statistical methods.

In LP based methods, deviations from the separating hyper-planes are used as measures of misclassification as mentioned above. On the other hand, the number of misclassifications can be considered directly in mixed integer linear programming (MILP) models in which binary variables are used to indicate whether instances are correctly or incorrectly classified. For two-group data classification problem, Bajgier and Hill [72] included the number of misclassifications and the deviations in the objective function of a

MILP model. On the other hand, Gehrlein [66] proposed a MILP approach for minimizing the number of misclassified instances in multi-group data classification problems, while Wilson [73] suggested an alternative MILP formulation and solution methods for these problems. Stam and Joachimsthaler [74] argued that these MILP based methods may be superior to both LP based techniques and statistical approaches. However, MILP approaches can be used to solve problems involving small number of instances due to computational reasons.

The problems that may appear in mathematical programming formulations for data classification are summarized by Koehler [75]. Specific problems include the choice of objective function, unacceptable or improper solutions, inconsistencies, gaps, and balancing of misclassifications. MP based data classification models must be normalized to prevent the generation of discriminant functions in which the variable coefficients and the constant term are zero. This normalization requirement can cause difficulties, and unlike statistical approaches, variables can not be selected in a computationally efficient way with MP models. Glen [76] developed two integer programming (IP) methods for normalizing MP discriminant analysis models. In the first method, binary variables are used to represent the constant term, but with this normalization functions with a zero constant term can not be generated. Moreover, the variable coefficients are not invariant under origin shifts. These limitations are overcome by the second method by using IP to constrain the sum of the absolute values of the variable coefficients to a constant [76]. Pavur and Loucopoulos [77] examined conditions under which degenerate solutions can occur in MILP models for the classification problem for more than two groups. They presented a multiple-group MSD model and a two-goal approach to the multiple-group data classification problem. Lam and Moy [78] proposed an aggregate model which simultaneously determines the cut-off values for the different classification functions in order to provide better estimates of the group boundaries.

Silva and Stam [79] introduced a computationally attractive algorithm, the Divide and Conquer (D&C), for determining classification rules which minimize the number of misclassifications in the training set for two-group data classification problems. The D&C algorithm partitioned the problem in smaller and more easily handled sub problems and solved the problem to the exact optimal solution by allowing analysis of much larger training sets than previous methods. On the other hand, Glen [80] developed an iterative MILP model to allow classification accuracy maximizing discriminant functions to be generated for problems with many more instances that can be considered by the standard MILP formulations. First, a discriminant function is generated by using a MSD based mathematical programming formulation for the complete set of instances. Then, a neighborhood of instances is defined and a MILP model is used to generate a discriminant function that maximizes classification accuracy within this neighborhood. This procedure is repeated until there is no improvement in the total number of instances classified correctly. This iterative MILP method is applied to a two-group classification problem involving 690 observations.

There are some very good MP based heuristics [81, 82] that can solve real world two-group data classification problems fast. Although there exist ways to solve a multi-group data classification problem by means of solving several two-group problems, such approaches bring about new problems [83]. Hence, Adem and Gochet [25] presented a MP based heuristic that avoid these problems and can tackle with multi-group data classification problems directly. The basic idea is to improve an LP-generated classifier with respect to the number of misclassifications on the design data set. The performance of the proposed approach is tested on both simulated and real world data sets.

In addition to the standard MP based data classification methods in which discriminant functions are generated by solving a single MP model, two-stage based MP methods have also been developed. Stam and Ragsdale [84] proposed a two-stage method which is particularly suitable for data classification problems with outlier contaminated data. In the first stage, a discriminant function is generated by solving the MSD based model. In that model, some of the instances could be misclassified. In the second stage, the objective is to generate a new discriminant function that minimizes a measure of total misclassification while ensuring that the correctly classified instances in the first stage remain correctly classified. Detailed information related to two-stage MP based methods and comparisons with standard MP based methods are given by Glen [85]. The results from comparisons of methods on one real data set and six simulated data configurations indicate that a single technique will not produce good linear classifier under all data conditions. Several methods should consider in developing classification models, with the most appropriate method chosen for a particular problem.

2.3 Literature Review on Protein Folding Type Prediction

Proteins are the molecules of life that play a key role in realizing the functions of any biological organism. Discovery of the functions of proteins will enable us to understand the principles of life and working mechanisms of any organism. In the case of humans, this discovery will lead to the design of new drugs that will regulate the functions of proteins in order to improve the quality of life. Functions of proteins are highly correlated to their three dimensional structure. There exist some experimental methods to determine the protein structure including *X*-ray diffraction and nuclear magnetic resonance (NMR). These experimental methods require long experimental times and large amounts of resources. In order to overcome these shortcomings of experimental methods, researchers have developed a host of methods to predict the protein structures. Due to the importance of protein structure in understanding the biological and chemical activities in any biological system, protein structure determination and prediction has been a focal research subject in computational biology and bioinformatics. The knowledge of folding type of proteins is an important part of protein structure prediction and determination studies. The results of the secondary structure prediction [86, 87] and the efficiency of searching the possible conformations of the tertiary structure [88, 89] could be significantly improved by incorporating the knowledge on folding types of protein. Another factor that motivates protein folding type prediction studies is the substantial gap between number of proteins for which structure is known and thus structural class can be assigned manually (approximately 30 000 proteins are stored in Protein Data Bank [90] and SCOP [91]) and the total number of currently known proteins (NCBI database contains over 2 million proteins). Therefore, development of a reliable method for prediction of folding types of proteins for new and undetermined protein sequences is very important.

A protein molecule is the chain(s) of amino acids (also called residues). There are 20 types of amino acids in nature and their names, three-letter representations and single-letter representations are provided in Table 2.1. Residue content and order in chain(s) is unique for each protein just like specificity of gene sequence.

Starting with the sequence of residues in the chain(s) making up protein, there are 4 basic structural phases: primary structure, secondary structure, tertiary structure and quaternary structure. The secondary structure (folding type) of a segment of polypeptide chain is the local spatial arrangement of its main-chain atoms without regard to the conformation of its side chains or to its relationship with other segments. This is the shape formed by amino acid sequences due to interactions between different parts of molecules. There are mainly three types of secondary structural shapes: α-helices, β-sheets and other structures connecting these such as loops, turns or coils. Alpha-helices are spiral strings formed by hydrogen bonds between CO and NH groups in residues. Beta-sheets are plain strands formed by stretched polypeptide backbone. When β-sheets come together, hydrogen bonds form between C=O and NH groups of residues of adjacent chains, keeping them together. Connecting structures do not have regular shapes; they connect α-helices and β-sheets to each other.

Table 2.1 List of amino acids, their three-letter and single-letter representations.

Amino Acid	Three Letter	Single Letter	Amino Acid	Three Letter	Single Letter
alanine	ALA	A	leucine	LEU	L
arginine	ARG	R	lysine	LYS	K
asparagine	ASN	N	methionine	MET	M
aspartic acid	ASP	D	phenylalanine	PHE	F
cysteine	CYS	C	proline	PRO	P
glutamic acid	GLU	Q	serine	SER	S
glutamine	GLN	E	threonine	THR	T
glycine	GLY	G	tryptophan	TRP	W
histidine	HIS	H	tyrosine	TYR	Y
isoleucine	ILE	I	valine	VAL	V

The proportion of α-helices and β-sheets in the secondary structures of proteins are used to determine the folding type of proteins. Protein folding type definitions were initially developed in 1980s and redefined multiple times since then (Table 2.2).

Table 2.2 Definitions of Protein Structural Classes.

Reference	Folding Type	Helix (α) amount	Strand (β) amount	Additional constraints
[92]	α proteins	>15%	<10%	
	β proteins	<15%	>10%	
	α+β proteins	>15%	>10%	Contains dominantly antiparallel β-sheets
	α/β proteins	>15%	>10%	Contains dominantly parallel β-sheets
	Irregular			Otherwise
[93]	α proteins	≥40%	≤5%	
	β proteins	≤5%	≥40%	
	α +β proteins	≥15%	≥15%	More than 60% antiparallel β-sheets
	α/β proteins	≥15%	≥15%	More than 60% parallel β-sheets
	Irregular	≤10%	≤10%	
[95]	α proteins	>15%	<10%	
	β proteins	<15%	>10%	
	Mixed proteins	>15%	>10%	
	Irregular			Otherwise
SCOP[91]	α proteins	NA	NA	Manual classification
	β proteins	NA	NA	Manual classification
	α+β proteins	NA	NA	Manual classification
	α/β proteins	NA	NA	Manual classification
	+7 other classes	NA	NA	Manual classification

The main differences were in the thresholds used to define amount of strands for all-α proteins, and amount of helices for all-β proteins. Nakashima and colleagues [92] defined five structural classes in 1986. Then, Chou [93] proposed classification into again five classes by using different thresholds in 1995. The change was due to Nakashima's classification, which set the thresholds for all-α proteins and all-β proteins that were not large enough to reflect the real features of the two structural classes. Chou also defined content of the secondary structures using the Dictionary of Secondary Structure of Proteins (DSSP) [94]. Eisenhaber and colleagues [95] proposed another definition which merges the α+β and the α/β classes into so-called mixed class and thus considers only four in 1996. In all above classifications, irregular proteins, ξ, are omitted from classification as they are small in numbers.

The threshold based classifications were replaced by the manually performed SCOP classification. The descriptions of the structural and evolutionary relationships of proteins from the Protein Data Bank (PDB) [90] are considered in the SCOP database [91]. The SCOP classifies proteins on multiple levels including structural classes, but also as belonging to different families, super families and containing different domains. Domain is defined as a structurally conserved part of a protein sequence, and together with the entire sequences is currently a target of structure prediction. The SCOP's classification does not incorporate hard coded rules for structural classes. Intuitively, it makes decisions based on structural elements that are located in individual domains that constitute the protein. Researchers claim that the SCOP classification is more "natural" and provides more reliable information to study protein structural classes when compared to classification based on the percentage amounts of the secondary structures [91, 96, 97]. The SCOP classification currently includes 11 classes [98]: (1) all-α proteins; (2) all-β proteins; (3) α/β proteins; (4) α+β proteins; (5) multi-domain proteins; (6) membrane and cell surface proteins; (7) small proteins; (8) coiled coils proteins; (9) low resolutions proteins; (10)

peptides; and (11) designed proteins. Usually, only the first four categories are considered for computational prediction purposes as they include significant majority of the protein sequences.

It is postulated that overall folding type of a protein depends on its amino acid composition [92]. There have been several methods proposed to exploit this postulate for predicting folding type of a protein. Chou [99] developed a new prediction algorithm which incorporates coupling effect between different amino acid components. By the help of this component-coupled algorithm, prediction quality was significantly improved. Another important progress in this area was achieved by Bahar *et al.* [89]. In their study, a compact lattice model was proposed in predicting structural class from amino acid composition and 81% accuracy achieved using singular value decomposition method [89]. In this method, each protein is represented by a 19-dimensional array of fluctuations in fractions of residues of different types. The j^{th} element of this vector is the difference between the composition of the amino acid type j and the average fraction of amino acid j in the group of n structures. The distance of a protein from the four type of structural classes are calculated using 19-dimensional array of the protein by applying singular value decomposition method. The smallest of the four distances obtained for each protein determines the structural class of that protein. Although they use the same data set and mathematically identical method with Chou, their accuracy is somehow less. They explore this puzzling difference and came up with the result that the data files used in these studies are different. Chou used files that contained fewer residues (chains of amino acids) compared with intact Protein Data Bank (PDB) files. Eisenhaber *et al.* [95] found that component coupling effect between amino acid components did not improve the class prediction, using a different dataset constructed according to their definition. In order to clarify this paradox, Zhou [100], Chou *et al.* [101] and Cai [102] showed that component-coupled algorithm significantly improved the prediction accuracy. The reasons why

Eisenhaber *et al.* come up with that result are misusing the component-coupled algorithm and using a conceptually incorrect rule to classify protein structural classes. On the other hand, Bu *et al.* [103] come up with a new idea, using amino acid index rather than composition in order to predict the structural classes. The overall predictive accuracy of the new proposed method for the jackknife test was 5-7% higher than the accuracy based only on the composition. However, many researchers continued studying on the first case, based on only the amino acid composition. Cai *et al.* [104] applied T. Kohonen's self-organization neural network on two data sets composed of 277 and 498 domains, respectively. They showed that this approach can be a powerful tool for protein structural class prediction. Furthermore, support vector machine (SVM) method was performed based on the same data sets by [100]. The SVM method applies for two class problems. Thus, "one-against other" method is used to transfer it into two class problems. Most recently, Kurgan and Homaeiang [23] provided a comprehensive literature survey and analyzed the impact of prediction algorithms and test procedures on accuracy. Consequently, the prediction of folding types from amino acid composition alone is an important topic, which has been the object of many recent researches. Existing data classification methods applied to protein folding type prediction is mainly appropriate for two-class problems. These methods can be modified for multi-class problems. Unfortunately, these modifications can cause the degradation of classification performance. Therefore, developed three-stage mathematical programming based hyper-box enclosure approach, which is capable of solving multi-class problems without any modification, can be used to classify a given primary protein structure into folding types according to its amino acid composition effectively.

In conclusion, there exists restricted number of methods for multi class data classification problems in literature. This thesis addresses the need for efficient and reliable methods for multi-class problems by introducing a new mixed-integer programming approach. Moreover, the important and widely used data sets, the protein folding type data set and UCI Repository data sets are studied to analyze the performance of the developed model. The results on these data sets show that the prediction accuracy of the developed model is as good as the existing data classification models in literature. Furthermore, developed model gets rid of some drawbacks of the available multi-class data classification models with only one adjustable parameter, rather short learning and computational time, no need to know the underlying distribution of the data and well-construction of the class boundaries.

Chapter 3

MILP BASED HYPER-BOX ENCLOSURE APPROACH

The objective in data classification is to assign instances that are described by several attributes into a predefined number of classes. The use of hyper-boxes for defining boundaries of the sets that include all or some of the instances in that set as shown in Figure 3.1 can be very accurate on multi-class problems. If it is necessary, more than one hyper-box could be used in order to represent a class as shown in Figure 3.1. When the classes that are indicated by square and triangle instances are both represented with a single hyper-box respectively, the boundaries of these hyper-boxes overlap. Thus, two boxes are constructed in order to eliminate this overlapping. A very important consideration in using hyper-boxes is the number of boxes used to define a class. If the total number of hyper-boxes is equal to the number of classes, then the data classification is very efficient. On the other hand; if there are as many hyper-boxes of a class as the number of instances in a class, then the data classification is inefficient.

The data classification problem is considered in two parts as training and testing. Determination of the characteristics of the instances that belong to a certain class and differentiating them from the instances that belong to other classes are the main objectives of the training part. The hyper-boxes that determine the characteristics of the classes are constructed in the training part by the help of mixed-integer linear programming (MILP) formulation. After the distinguishing characteristics of the classes are determined, then the effectiveness of the classification is observed by the help of distance-based testing algorithm. Predictive accuracy of the developed model is performed on a test data set during the test part.

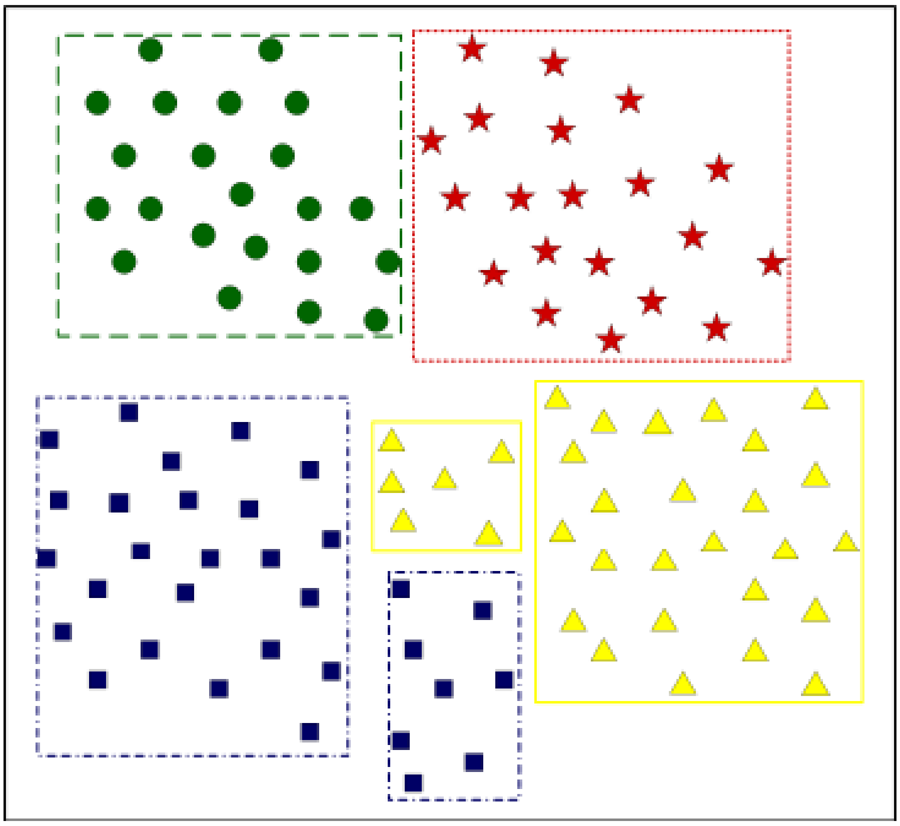

Figure 3.1 Schematic representation of multi-class data classification using hyper-boxes.

3.1 Training Algorithm: MILP Formulation

Training part studies are performed on a training data set composed of a number of instances i. The instances are represented by the parameter a_{im} that denotes the value of attribute m for the instance i. The class k that the instance i belongs to are given by the set D_{ik}. Each existing hyper-box l encloses a number of instances belonging to the class k. Moreover, bounds n (*lower, upper)* of each hyper-box is determined by solving the training problem. M and N represents the total number of attributes and bounds, respectively.

Given these parameters and the sets, the following binary and continuous variables are sufficient to model the data classification problem with hyper-boxes. The existence of hyper-box l is represented by binary variable yb_l. The binary variable ypb_{il} indicates the position (inside or outside) of the instance i with respect to box l. The binary variables ybc_{lk} and ypc_{ik} indicate the assigned class k of instance i and hyper-box l, respectively. If the instance i is within the bound n with respect to attribute m of hyper-box l, then the binary variable $ypbn_{ilmn}$ is 1, otherwise 0. Similarly, $ypbm_{ilm}$ indicates whether the instance i is within the bounds of attribute m of hyper-box l or not. Finally, yp_{ik} indicate the misclassification of instance i to class k. In order to define the boundaries of hyper-boxes, two continuous variables are required: X_{lmn} is the one that models bounds n for box l on attribute m. Correspondingly, bounds n for box l of class k on attribute m are defined with the continuous variable XD_{lkmn}.

The following MILP problem models the training part of data classification method using hyper-boxes:

$$\min \quad z = \sum_i \sum_k yp_{ik} + \sum_l yb_l \tag{3.1}$$

subject to

$$XD_{lkmn} \leq a_{im} ypb_{il} + Q(1 - ypb_{il}) \quad \forall i,k,l,m,n \mid n = lower \tag{3.2}$$

$$XD_{lkmn} \geq a_{im} ypb_{il} \quad \forall i,k,l,m,n \mid n = upper \tag{3.3}$$

$$XD_{lkmn} \leq Q ybc_{lk} \quad \forall k,l,m,n \tag{3.4}$$

$$\sum_k XD_{lkmn} = X_{lmn} \quad \forall l,m,n \tag{3.5}$$

$$ypbn_{ilmn} \geq \frac{1}{Q}(X_{lmn} - a_{im}) \quad \forall i,l,m,n|n = upper \tag{3.6}$$

$$ypbn_{ilmn} \geq \frac{1}{Q}(a_{im} - X_{lmn}) \quad \forall i,l,m,n|n = lower \tag{3.7}$$

$$\sum_{l} ypb_{il} = 1 \quad \forall i \tag{3.8}$$

$$\sum_{k} ypc_{ik} = 1 \quad \forall i \tag{3.9}$$

$$\sum_{l} ypb_{il} = \sum_{l} ypc_{ik} \quad \forall i \tag{3.10}$$

$$\sum_{k} ybc_{lk} = yb_{l} \quad \forall l \tag{3.11}$$

$$ybc_{lk} \le \sum_{i} ypb_{il} \quad \forall l,k \tag{3.12}$$

$$ybc_{lk} \le \sum_{i} ypc_{ik} \quad \forall l,k \tag{3.13}$$

$$\sum_{n} ypbn_{ilmn} - ypbm_{ilm} \le N - 1 \quad \forall i,l,m \tag{3.14}$$

$$\sum_{m} ypbm_{ilm} - ypc_{ik} \le M - 1 \quad \forall i,l,k \tag{3.15}$$

$$ypc_{ik} \le yp_{ik} \quad \forall i,k \notin D_{ik} \tag{3.16}$$

$$X_{\mathrm{lmn}}, XD_{\mathrm{lkmn}} \ge 0 \tag{3.17}$$

$$yb_{l}, ypb_{il}, ypc_{ik}, ybc_{lk}, ypbn_{ilmn}, ypbm_{ilm}, yp_{ik} \in \{0,1\} \tag{3.18}$$

Minimization of the misclassified instances in the data set with the minimum number of hyper-boxes is the objective of the MILP model given in (3.1). The lower and upper bounds of the hyper-boxes are determined by the instances that are enclosed within the hyper-boxes. Hence, lower and upper bounds of hyper-boxes are calculated by equations (3.2) and (3.3), respectively. Eq. (3.4) enforces the bounds of hyper-boxes exist if and only if this hyper-box is assigned to a class. The relationship between two continuous variables is given in Eq. (3.5). The position of an instance with respect to the bounds on attribute *m* for a hyper-box is given in Eqs. (3.6) and (3.7). The binary variable $ypbn_{ilmn}$ helps to identify whether the instance *i* is within the hyper-box *l*. Two constraints, one for the lower bound and one for the upper bound, are needed for this purpose (Eqs. (3.6) and (3.7)). Since these constraints establish a relation between continuous and binary variables, a large parameter, Q, is included. Q generally takes the maximum attribute value in the data set. The assignment of an instance to a single hyper-box l and a single class k is established by the equations (3.8) and (3.9), respectively. The equivalence between Eqs. (3.8) and (3.9) is given in Eq. (3.10); indicating that if there is an instance in the class k, then there must be a hyper-box l to represent the class k and vice versa. The existence of a hyper-box implies the assignment of that hyper-box to a class as shown in Eq. (3.11). If a class is represented by a hyper-box, there must be at least one instance within that hyper-box as in Eq. (3.12). In the same manner, if a hyper-box represents a class, there must be at least an instance within that class as given in Eq. (3.13). The Eq. (3.14) represents the condition of an instance being within the bounds of a box in attribute m. If an instance is within the bounds of all attributes of a box, then it must be in the box as shown in Eq. (3.15). When an instance is assigned to a class that it is not a member of, a penalty applies as indicated in Eq. (3.16). Finally, last two constraints Eq. (3.17) and (3.18) give non-negativity and integrality of decision variables. The model has $LMN + LKMN$ continuous variables, $L + LK + 3IK + IL + ILMN + ILM$ binary variables and $O(IKLM)$ constraints.

3.2 Three-Stage Approach

Solving the proposed MILP problem to optimality is computationally expensive for large multi-group data classification problems. The major source of computational difficulty is the potentially large number of binary variables. Hence, we propose a three-stage decomposition algorithm (shown in Figure 3.2) for obtaining optimal solutions to MILP model.

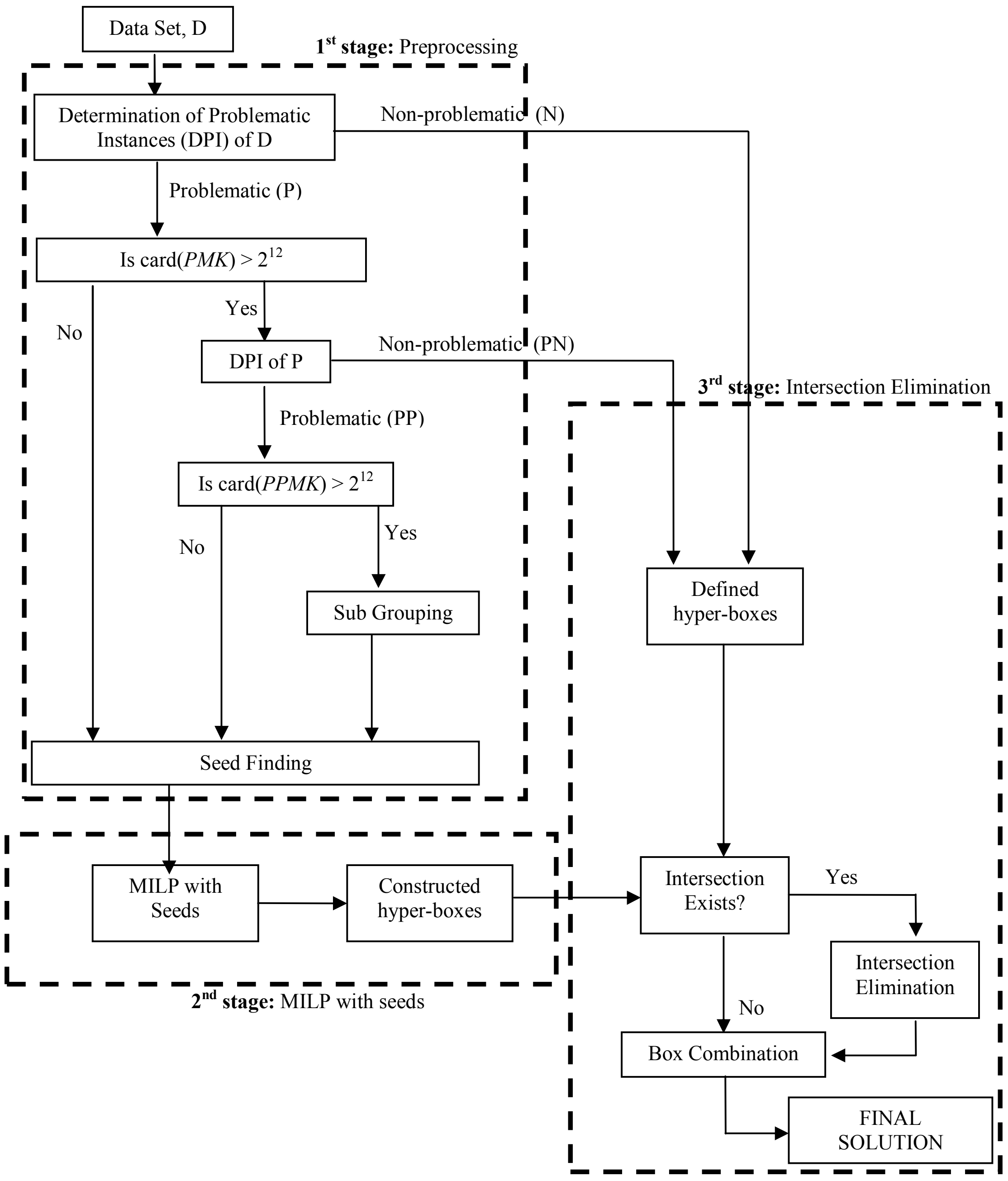

Figure 3.2 Flowchart of the decomposition algorithm for solving multi-class classification algorithm using hyper-boxes.

Instances that are difficult to classify are identified in the first stage that is referred to as preprocessing. Moreover, sub grouping and seed finding algorithms are applied to improve the computational efficiency. With greater emphasis given to these observations, solution to the problem is obtained in the second stage using the MILP formulation. Last, final assignments, elimination of box intersections and box combination procedures are carried out in the third step.

3.2.1 Preprocessing

First, maximum and minimum attribute values for each class are determined. Then, the boundaries of the classes are compared to check whether they overlap or not. If the boundaries of the classes overlap, then the instances that are enclosed by other classes are identified. These instances are called as 'problematic' instances, since they are not separable from the instances of the other classes with a single hyper-box. In the case of having large number of 'problematic' instances, the same procedure is repeated to reduce the total number of such instances. In some cases, applying one or two times the same procedure do not reduce the number of problematic instances as we want. For those cases, we proposed a sub grouping algorithm in order to obtain small sub groups from the data sets efficiently.

The proposed MILP model has $O(LKMN)$ continuous variables, $O(ILMN)$ binary variables and $O(IKLM)$ constraints. For each instance removed in the preprocessing step, the binary variables and constraints in the MILP model are reduced by $O(LMN)$ and $O(KLM)$, respectively.

3.2.2 Threshold Value for the Number of Problematic Instances

In order to give more formal threshold value for the number of problematic instances, we perform some runs with different number of instances. For this purpose, sub problems of a protein folding type data set are used. By increasing the number of instances, we try to observe the CPU times of the runs with respect to the change in the size of the problem (Figure 3.3.). For each problem size, we perform 10 different runs and give the average results. In this graph, *I* represents the number of instances, *K* represents the number of classes and *M* represents the number of attributes. The problem size is given by the products of cardinalities of *I*, *M* and *K*. As this product increases, the number of binary and continuous variables in the MILP model increases. Thus, the required solution time increases by the increase in the problem size. After some point, this increase is much more significant. As it could be observed from the graph, the threshold value is 2^{12} (4096). After card(*IMK*) achieves that value, the required CPU time is high and unfavorable.

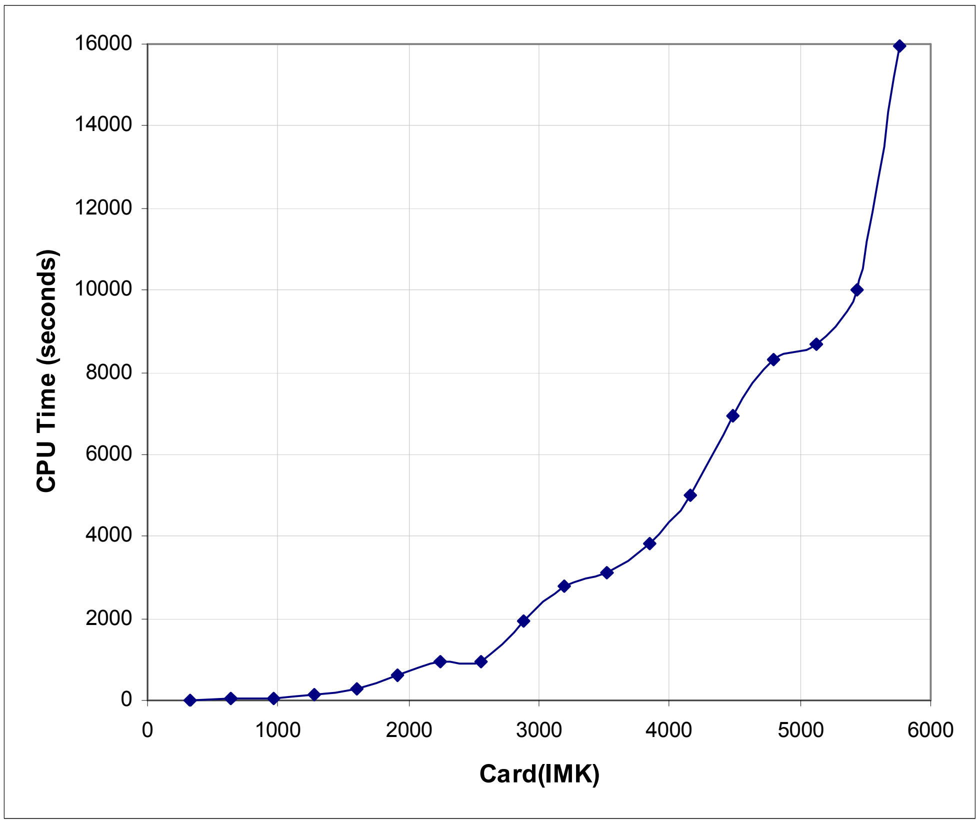

Figure 3.3 Problem size versus CPU time of algorithm.

3.2.3 Sub Grouping Algorithm

For some of the data classification problems, the number of problematic instances is so high that this step does not make enough improvement in the computational time of the given problem. Hence, for this type of problems a sub grouping algorithm is proposed in order to improve the computational efficiency. Sub grouping is a method that constitutes a given number of subsets of the given data set by selecting instances considering some similarity-dissimilarity measure.

The determination of subsets is crucial: the instances for each subset must be chosen to ensure that they are separated well from other instances. We develop a pure integer programming (IP) formulation to accomplish this task.

As in the MILP, instances are represented by the parameter a_{im} that denotes the value of attribute *m* for the instances *i*. The class *k* of instance *i* belongs to is given by the set D_{ik}. NI_k represents the number of instances in class *k*. Moreover, $DB_{ii'}$ represents the distance between two data points *i* and *i'*. This distance is calculated using Euclidean distance in *m*-dimensional space as given in Equation (3.19).

$$DB_{ii'} = \sqrt{\sum_m (a_{im} - a_{i'm})^2} \tag{3.19}$$

Given these parameters and the sets, the similarity, S_i, and dissimilarity, DS_i, of an instance *i* can be calculated as in Equations (3.20) and (3.21), respectively. Similarity, S_i, is the average distance from instance *i* to instances *i'* that exist in the same class with instance *i*. On the other hand, dissimilarity, DS_i, is the average distance from instance *i* to instances *i'* that are not in the same class with instance *i*.

$$S_i = \frac{\sum_{i' \in D_{ik}} DB_{ii'}}{NI_{k:i \in D_{ik}}} \tag{3.20}$$

$$DS_i = \frac{\sum_{i' \notin D_{ik}} DB_{ii'}}{\left(\sum_k NI_k\right) - NI_{k:i \in D_{ik}}} \tag{3.21}$$

The binary variable *SP_i*, that indicates whether the instances *i* is selected for this sub group or not, is sufficient to model sub grouping problem. Furthermore, *SS* is the number of instances that exist in each of the constructed sub groups from the given data set *D*. *TS* is the number of sub groups that should be obtained. *TS* and *SS* can be determined by using the Equations (3.22) and (3.23).

$$TS = \left\lceil \frac{card(DMK)}{2^{12}} \right\rceil \tag{3.22}$$

$$SS = \frac{card(D)}{TS} \tag{3.23}$$

The following IP-Sub Group models the sub grouping problem and select *SS* number of instances to form a sub group:

IP-Sub Group:

$$\min\ z = \sum_i SP_i \left(S_i - DS_i\right) \tag{3.24}$$

subject to

$$\sum_i SP_i = SS \tag{3.25}$$

$$SP_i \in \{0,1\} \quad \forall i \tag{3.26}$$

The objective of the IP-Sub Group problem given in Eq. (3.24) is to minimize the similarities measures and maximize the dissimilarities measures of selected instances. Equation (3.25) states that the number of selected instances must be exactly *SS*. Finally, integrality of the decision variable SP_i is given by (3.26).

This IP-Sub Group model constitutes a single subset, S_l, from a given data set D. In order to obtain each subset, one should solve TS-1 consecutive IP-Sub Group model while in each case updating the new dataset D_{new} as $D_{old} \backslash S_i$. Hence, by solving IP-Sub Group models, we will obtain TS sub groups of data set D. As MILP is based on hyper-boxes approach, this sub group decomposition will not affect the inherent properties of this approach. Moreover, sub grouping will improve the computational efficiency of the overall data classification method.

Further investigation on the proposed IP-Sub Group model leads us to the following property.

Property 3.1: Total Unimodularity Property [105]

Let A be an mxn integer matrix with a rank of m. A is unimodular if the determinant of every basis matrix B of A has value +1 or -1 as given by Ahuja *et al.* [105]. Thus, relying on this, we can state that if an integer valued matrix A is unimodular, then every basic feasible solution of the polyhedron defined by the constraints Ax = b where $x \geq 0$, is integer for every integer valued right hand side vector b. If every square submatrix of A has a determinant of 0 or ±1, then the matrix A is totally unimodular. Moreover, every totally unimodular matrix is unimodular since each basis matrix B of the matrix A has a determinant ±1 [105].

Proposition 3.1: The constraint set of the IP-Sub Group model has the total unimodularity property.

Proof: For the equation 3.25, I is the total number of instances. The corresponding A matrix of the IP-Sub Group model can be stated algebraically as follows.

$$\begin{array}{cccccccc} & SP_1 & SP_2 & \ldots & SP_i \ldots & & SP_{I-1} & SP_I \\ A = [& 1 & 1 & \cdots & 1 & \cdots & 1 & 1] \end{array}$$

The rank of above 1*xI* matrix is equal to 1 since it consists of only one row. Moreover, every square submatrix of *A* has a determinant +1 and therefore it is a totally unimodular matrix. Thus, IP-Sub Group model has the total Unimodularity property. □

Using this property, we can conclude that every basic feasible solution of the LP relaxation of IP-Sub Group model defined by Equation 3.25 is integer. Therefore, optimal solution of LP-relaxation is the optimal solution of IP-Sub Group model which means that solution of IP-Sub Group model could be easily obtained in a small amount of time.

In order to clarify the sub grouping approach, we tested IP-Sub Group model on an illustrative example given in Figure 3.4. In this illustrative example, there exist 100 instances (25 from each of the four classes) represented by two attributes values.

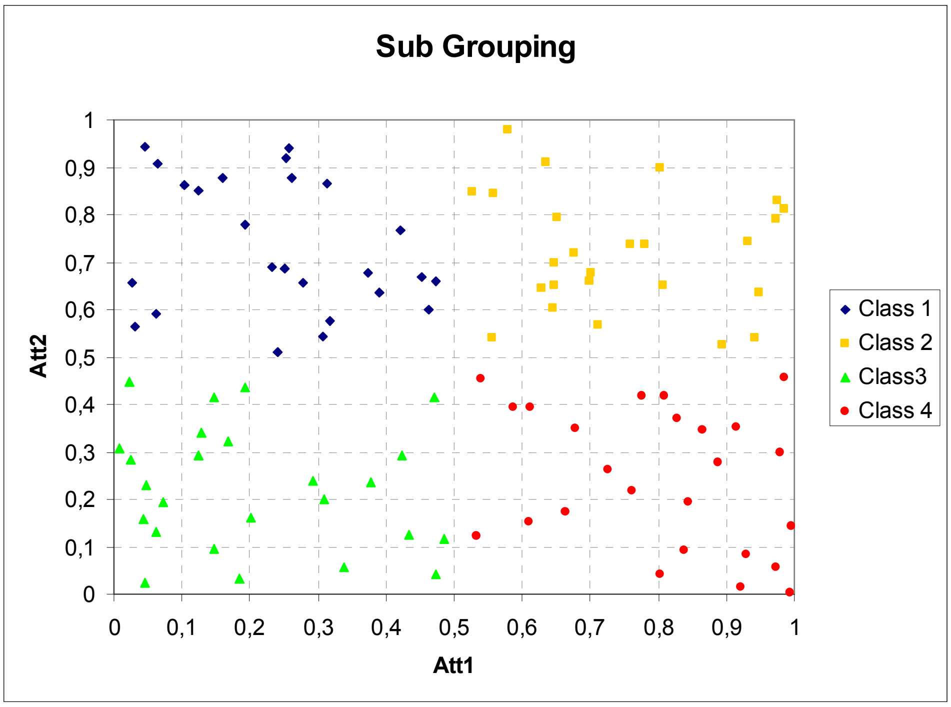

Figure 3.4 Illustrative example for sub grouping algorithm.

When the IP-Sub Group problem is solved for this illustrative example, we constitute two sub groups with 50 instances. The resulting sub groups are shown in Figures 3.5 and 3.6. As it can be seen from obtained sub groups, IP-Sub Group model efficiently selects the instances and constitute easier sub problems for MILP model. Solving the overall problem takes much more computational time with respect to solving two sub group problems separately. Hence, by solving Sub Group 1 and Sub Group 2 instances one by one using MILP, we obtain the constructed hyper-boxes in a reasonable amount of time. In some cases, obtaining the optimal solution of the overall problem takes more than a week/month. Therefore, in those cases solving the IP-Sub Group model and decompose the overall problem into smaller sub groups is favorable and preferable.

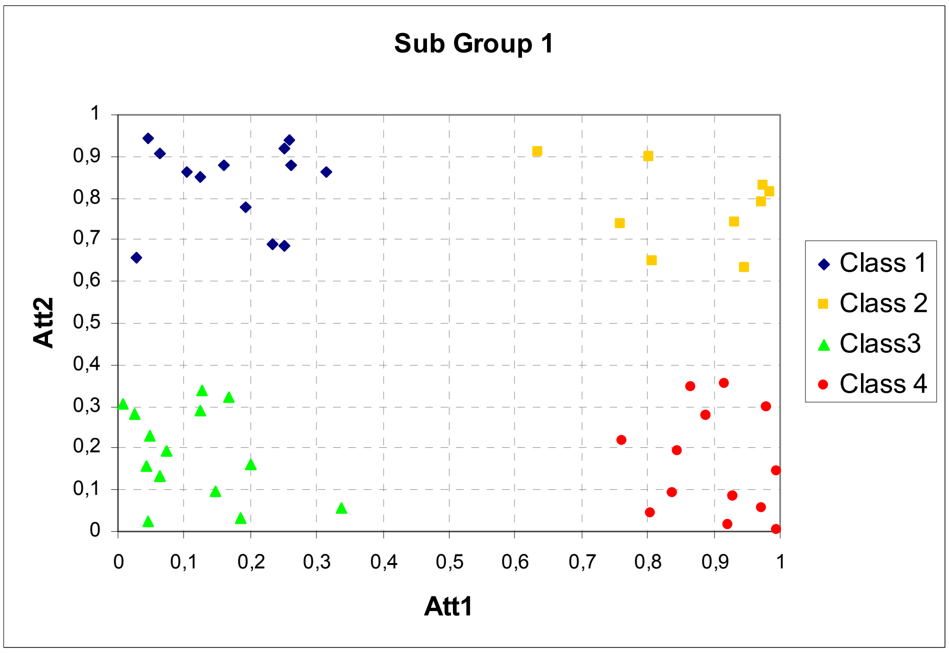

Figure 3.5 Sub Group 1 of given illustrative example.

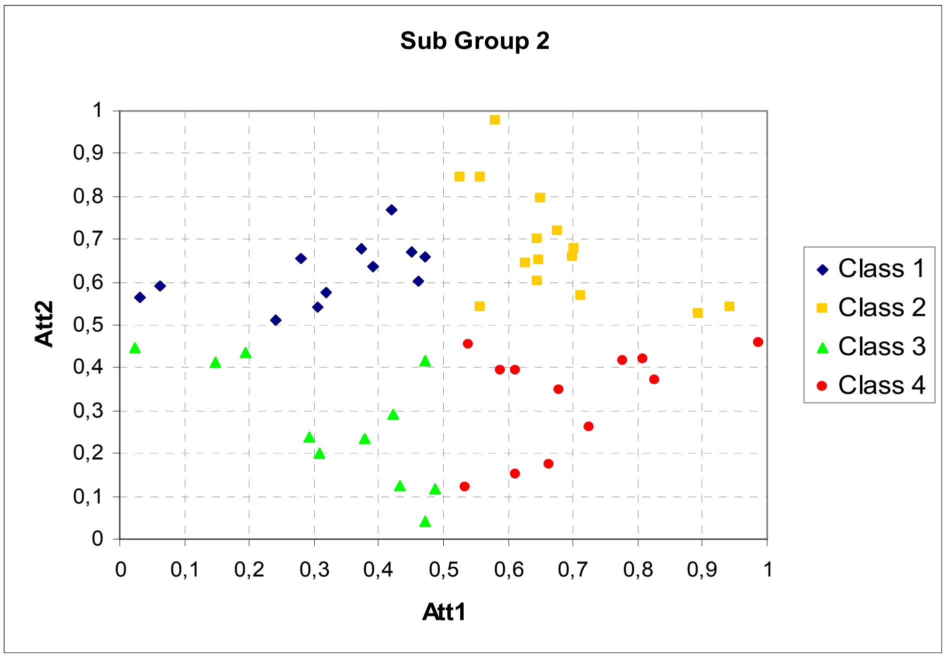

Figure 3.6 Sub Group 2 of given illustrative example.

3.2.4 Seed Finding Algorithm

Another method to improve the computational efficiency is determining representative seeds for each class. Seed finding is a method that selects an instance (seed) for each class and fixes assignments of these instances to their respective classes before solving the problem. The seeds improve the computational performance of the model without changing the optimal solution.

The determination of seeds is a critical task: the seeds for each class must be chosen to ensure that seeds are separated well from each other as well as being a good example of the group of instances in the same class. We develop a pure integer programming (IP) formulation to accomplish this task. As in the MILP formulation, instances are represented by the parameter a_{im} that denotes the value of attribute m for the instances i. The class k of instance i belongs to is given by the set D_{ik}. Moreover, $PP_{ii'}$ represents the distance between two instances *i* and *i'*. This distance is calculated using Euclidean distance in *m*-dimensional space as given in Equation (3.27).

$$PP_{ii'} = \sqrt{\sum_{m} (a_{im} - a_{i'm})^2} \tag{3.27}$$

Given these parameters and the sets, the binary variable YP_i, that indicates whether the instance *i* is selected as seed or not, is sufficient to model the seed finding problem. The following IP-Seed models the seed finding problem:

IP-Seed: $$\min z = \sum_{k} \sum_{i \in k} \sum_{i' \in k} PP_{ii'} YP_i - \left(\frac{1}{card(i \in k)} \right) * \sum_{k} \sum_{i \in k} \sum_{i' \notin k} PP_{ii'} YP_i \tag{3.28}$$

subject to

$$\sum_{i \in k} YP_i = 1 \quad \forall k \tag{3.29}$$

$$YP_i \in \{0,1\} \quad \forall i \tag{3.30}$$

The objective of the IP-Seed problem given in Eq. (3.28) is to minimize the distances from each seed to instance of its group (in-class distances) and maximize the average distances from each seed to the instances that belong to other classes (out-class distances). Equation (3.29) states that every class must have exactly one seed. Finally, integrality of the decision variable YP_i is given by (3.30).

We performed a set of experiments on MILP model without seeds to compare its results with the one initiated with seeds. One can observe the positive effect of seed finding algorithm on the solution of MILP model, in terms of improvement in the number of iterations, the number of nodes and the CPU times required to construct the hyper-boxes by comparing the results given in Table 3.1. In Table 3.1, *i* is the number of instances, *Cons.* is the number of constraints, *BVar* is the number of binary variables and *CVar* is the number of continuous variables in the model. When we analyze the Table 3.1, we see that

CPU times, number of iterations and nodes decrease significantly as introducing seeds to the model. Hence, seed finding algorithm improves the computational time requirement of the MILP model.

Table 3.1 A comparison of MILP model with and without seeds.

Problem Characteristics				**MILP without seeds**			**MILP with seeds**		
i	**# of Cons.**	**# of BVar**	**# of CVar**	**# of Iterations**	**# of nodes**	**CPU (sec.)**	**# of Iterations**	**# of nodes**	**CPU (sec.)**
10	12,265	6,190	2,081	57,543	331	81.14	15	0	0.468
20	22,435	12,330	2,161	114,470	239	458.843	1,152	0	2.296
30	32,605	18,470	2,241	187,769	603	1062.90	3,467	10	3.796
40	42,775	24,610	2,321	297,133	350	2154.35	26,390	270	27.593
50	52,945	30,750	2,401	432,922	862	4786.1	22,945	283	29.343

The seeds found by IP-Seed model are given in Figure 3.7. As it can be observed, seeds found by IP-Seed well exemplify the class properties.

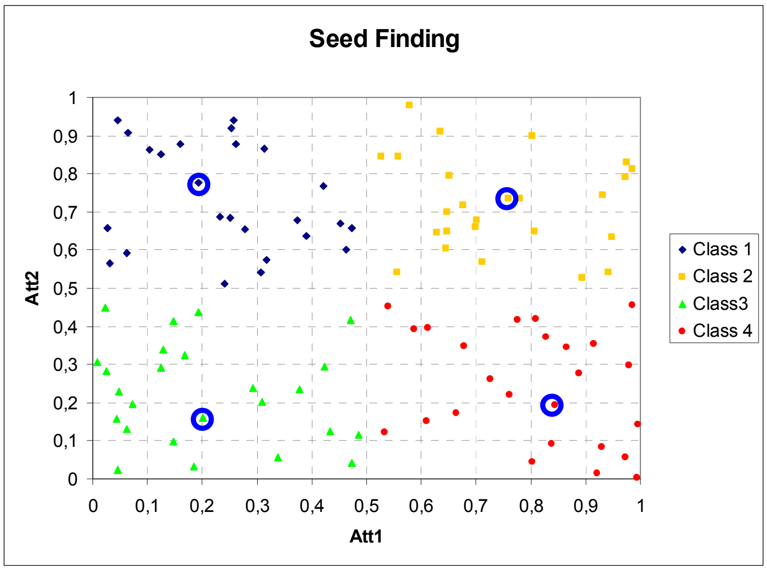

Figure 3.7 Seeds found by IP-Seed are circled on an illustrative example.

Further investigation on the proposed IP-Seed model leads us to the following property.

Proposition 3.2: The constraint set of the IP-Seed model has the total unimodularity property.

Proof: For the Equation 3.29, I is the total number of instances, K is the total number of classes, c is the total number of class 1 instances and t is the total number of class 2 instances. The corresponding A matrix of the IP-Seed model can be stated algebraically as follows.

$$A = \begin{array}{c|cccccccccccc|} & YP_1 & YP_2 & \ldots & YP_c & YP_{c+1} & YP_{c+2} & \ldots & YP_{c+t} & \ldots & \ldots & YP_{I-1} & YP_I \\ 1 & 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 & \cdots & \cdots & 0 & 0 \\ 2 & 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ K & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & \cdots & 1 & 1 \end{array}$$

The above KxI matrix is 0-1 matrix and its rank is equal to n since it is consists of n linearly independent rows. Moreover, every square submatrix of A has a determinant 0 or +1 and therefore it is a totally unimodular matrix. Thus, IP-Seed model has the total Unimodularity property. □

By the help of this property, we can conclude every basic feasible solution of the LP relaxation of IP-Seed model defined by Constraint 3.29 is integer. Therefore, optimal solution of LP-relaxation is the optimal solution of IP-Seed model which means that solution of IP-Seed model could be easily obtained in a small amount of time.

3.2.5 MILP Model

Once the *k* seeds to be assigned to the *k* classes are determined by IP-Seed model, we can solve MILP model for ‘problematic instances’ with these seeds. Assignment of the instances selected as seed in the MILP model means that we are setting the variables corresponding to these instances to a specific value. Hence, optimal values for associated variables are given and do not need to be optimized. This means that, some of the solutions in the solution space are eliminated by fixing these values. Thus, this approach is capable of obtaining alternative optimal solutions for MILP model with smaller computational effort.

3.2.6 Final assignment and Intersection Elimination

Since the MILP model is solved for ‘problematic instances’ only, the ‘non-problematic instances’ are assigned to hyper-boxes in a straight forward way. We define *k* hyper-boxes for each class and assign a ‘non-problematic instance’ to corresponding newly defined hyper-box. Each ‘non-problematic instance’ is considered one by one until all of these instances are assigned to a hyper-box. Finally, the bounds of these new hyper-boxes are determined by considering the maximum and minimum attribute values of all instances in these hyper-boxes. It is possible that these hyper-boxes have intersections. Instances are separated from the original hyper-box until all intersections are eliminated. The eliminated instances are grouped in a new box and intersection checking and elimination procedure is repeated until no more intersections occur between all of the constructed and defined hyper-boxes. After intersection elimination, box combination algorithm is included in order to get tight hyper-boxes for each class.

3.2.7 Box Combination

Box combination is the last step in the three-stage hyper-box enclosure approach. Since we do not solve problematic and non-problematic instances together, we could have some hyper-boxes that could be combined without causing any intersection. As we want to differentiate the class boundaries with minimum number of hyper-boxes, combination of these hyper-boxes and decreasing the number of overall hyper-boxes is preferable. Hence, we developed an integer programming (IP) formulation to accomplish this task. As in the MILP formulation, X_{lmn} represents the bounds of existing hyper-boxes or the hyper-boxes obtained at the end of intersection elimination algorithm. The index l represents the existing hyper-boxes and the index l' represents the hyper-boxes that are obtained by combinations of the existing ones. The class k of hyper-box l belongs to is given by the set BC_{lk}. $NX_{l'mn}$ represents the bounds of hyper-boxes l' that is obtained by combining the existing hyper-boxes that are in the same class. The class k of hyper-box l' belongs to is given by the set $NBC_{l'k}$. In order to define the box intersections, we need to use center and length of the hyper-boxes. The centers C_{lm} and $C_{l'm}$ can be calculated using the Equations (3.31) and (3.32), respectively. The lengths L_{lm} and $L_{l'm}$ can be calculated using the Equations (3.33) and (3.34), respectively. If the difference between the centers of the hyper-boxes is greater than the average lengths of the hyper-boxes for an attribute, then there is no intersection between these hyper-boxes for that attribute. Otherwise, these hyper-boxes will intersect on that attribute (Figure 3.8).

$$C_{lm} = \frac{X_{lmn|n=upper} + X_{lmn|n=lower}}{2} \tag{3.31}$$

$$C_{l'm} = \frac{NX_{l'mn|n=upper} + NX_{l'mn|n=lower}}{2} \tag{3.32}$$

$$L_{lm} = X_{lmn|n=upper} - X_{lmn|n=lower} \tag{3.33}$$

$$L_{l'm} = NX_{l'mn|n=upper} - NX_{l'mn|n=lower} \tag{3.34}$$

Given these parameters and the sets, the binary variables $IN1_{ll'm}$ and $IN2_{ll'm}$ are necessary to indicate the intersection of hyper-boxes l and l' for each attribute m. $IO_{ll'}$ is a binary variable that represents the intersection of hyper-box *l* and *l'*. The binary variable $CO_{l'}$ is 1 if there is an intersection related to newly defined hyper-box *l'*. Finally, $SO_{l'}$ is a binary variable which takes the value 1 when the hyper-box *l'* could be obtained without causing any intersection. The parameter $SI_{ll'}$ is 1 if the hyper-box *l'* is not obtained by any combination of the hyper-box *l* with other hyper-boxes and 0 otherwise. This parameter is necessary to check intersection for only the rest of the hyper-boxes that are not combined. Furthermore, the parameter is $SN_{l'l''}$ is 1 if hyper-box *l'* and hyper-box *l''* is obtained by combination of a common hyper-box and 0 otherwise. This parameter is necessary to eliminate the multiple selections of hyper-box *l* for combination.

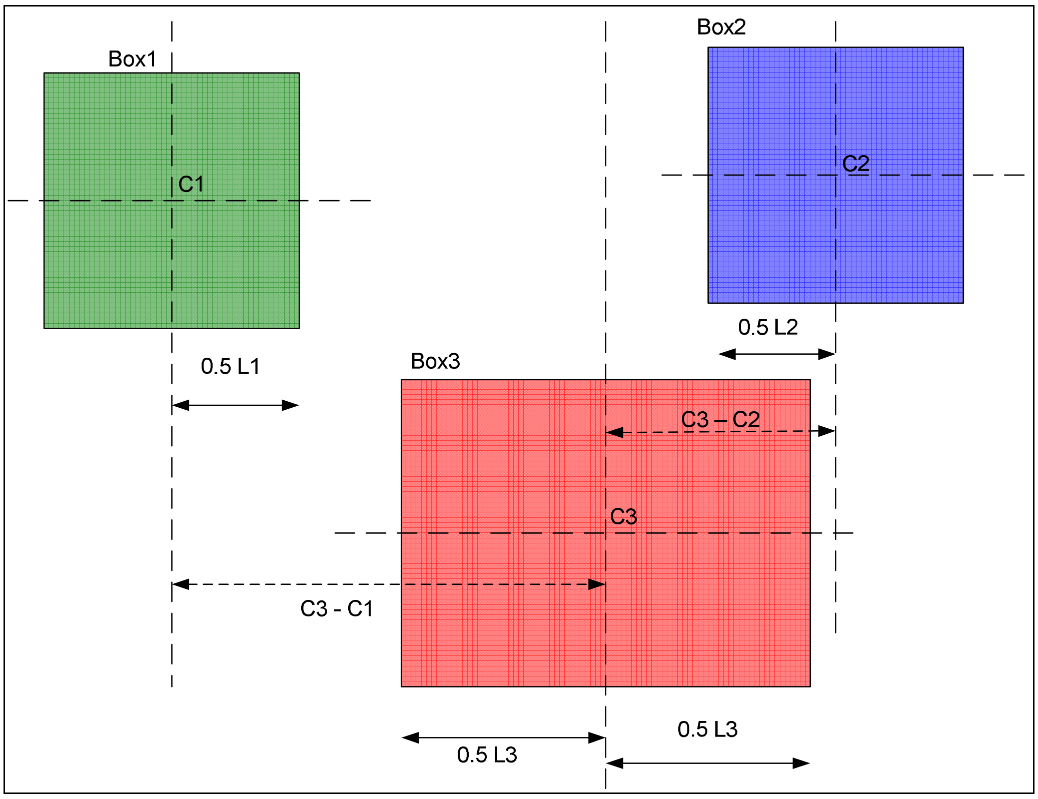

Figure 3.8 Hyper-box intersection check via the centers and lengths of hyper-boxes.

Using these binary variables and the parameters, the following IP-Box Combine models the box combination problem:

IP-Box Combine:

$$\max\ z = \sum_{l'} SO_{l'} \tag{3.35}$$

subject to

$$C_{l'm} - C_{lm} + Q \cdot IN1_{ll'm} \geq \frac{L_{lm} + L_{l'm}}{2} + \varepsilon \qquad \forall l, l', m \mid SI_{ll'} = 1 \tag{3.36}$$

$$C_{lm} - C_{l'm} + Q \cdot IN2_{ll'm} \geq \frac{L_{lm} + L_{l'm}}{2} + \varepsilon \qquad \forall l, l', m \mid SI_{ll'} = 1 \tag{3.37}$$

$$\sum_{m} (IN1_{ll'm} + IN2_{ll'm}) - 2 * card(m) + 1 \leq IO_{ll'} \qquad \forall l, l' \mid SI_{ll'} = 1 \tag{3.38}$$

$$IO_{ll'} \leq CO_{l'} \qquad \forall l, l' \mid SI_{ll'} = 1 \tag{3.39}$$

$$CO_{l'} + SO_{l'} \leq 1 \qquad \forall l' \tag{3.40}$$

$$SO_{l'} + SO_{l''} \leq 1 \qquad \forall l', l'' \mid SN_{l'l''} = 1 \tag{3.41}$$

$$SO_{l'}, CO_{l'}, IO_{ll'}, IN1_{ll'm}, IN2_{ll'm} \in \{0,1\} \qquad \forall l, l', m \tag{3.42}$$

The objective of the IP-Box Combine problem given in Eq. (3.35) is to maximize the number of newly obtained hyper-boxes that represents the combination of old ones. Equation (3.36) and (3.37) are necessary to count the intersections of existing and newly obtained hyper-boxes for an attribute. In order to give the relationship between the centers and lengths and intersections, a large parameter *Q* and ε are included in these constraints. If hyper-boxes intersect for all of the attributes, then the binary variable $IO_{ll'}$ is 1 with Equation (3.38). If newly-obtained hyper-box *l'* has any intersection with existing ones, then the corresponding binary variable $CO_{l'}$ will be 1 to represent the infeasibility of obtaining hyper-box l' (3.39). If obtaining the hyper-box l' is feasible, then the binary variable $SO_{l'}$ is 1, and 0 otherwise by Equation (3.40). The Equation (3.41) states that only one combination related to hyper-box l could be selected. Finally, integrality of the decision variables is given by (3.42).

The IP-Box Combine model tries to find the maximum number of hyper-box combinations and obtain combined hyper-boxes. It is not possible to get all of the hyper-box combinations after a single run. We should iteratively solve IP-Box Combine model until the objective function value is 0. In Figure 3.9, there is an artificial example to observe the behaviors of the IP-Box Combine model. After the first run of IP-Box Combine model, some of the hyper-boxes are combined but there are some more feasible combinations (Figure 3.10). After the second run of IP-Box Combine, all of the feasible combinations are obtained (Figure 3.11).

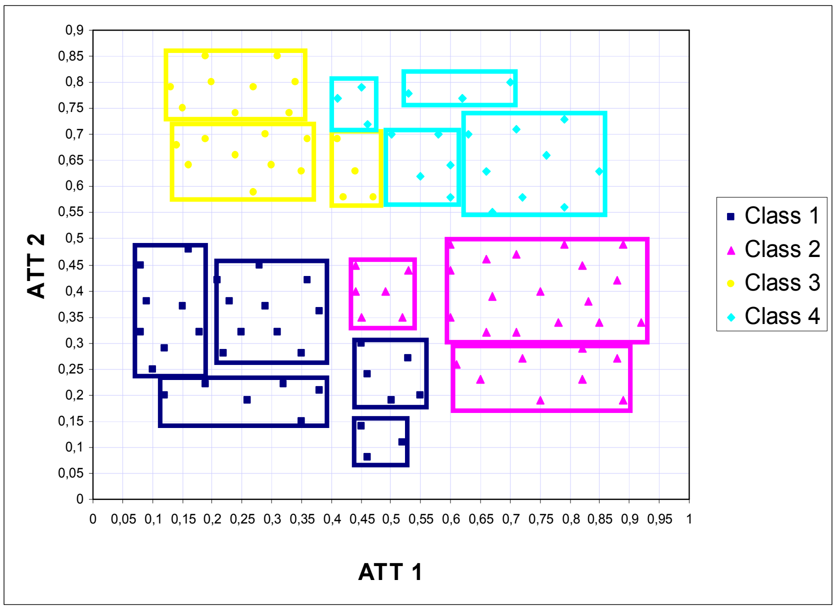

Figure 3.9 Artificial example for IP-Box Combine analysis.

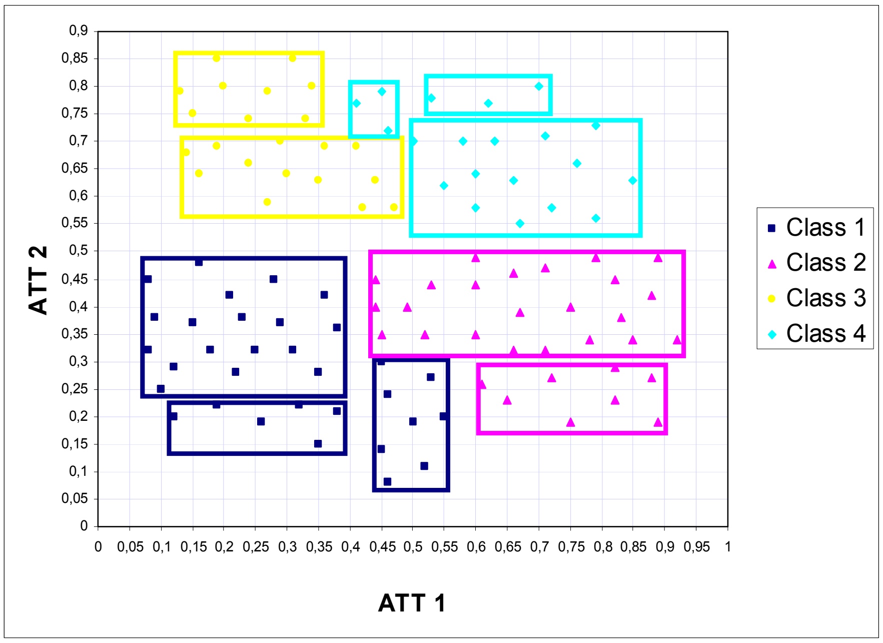

Figure 3.10 Combined hyper-boxes after the first run of IP-Box Combine model.

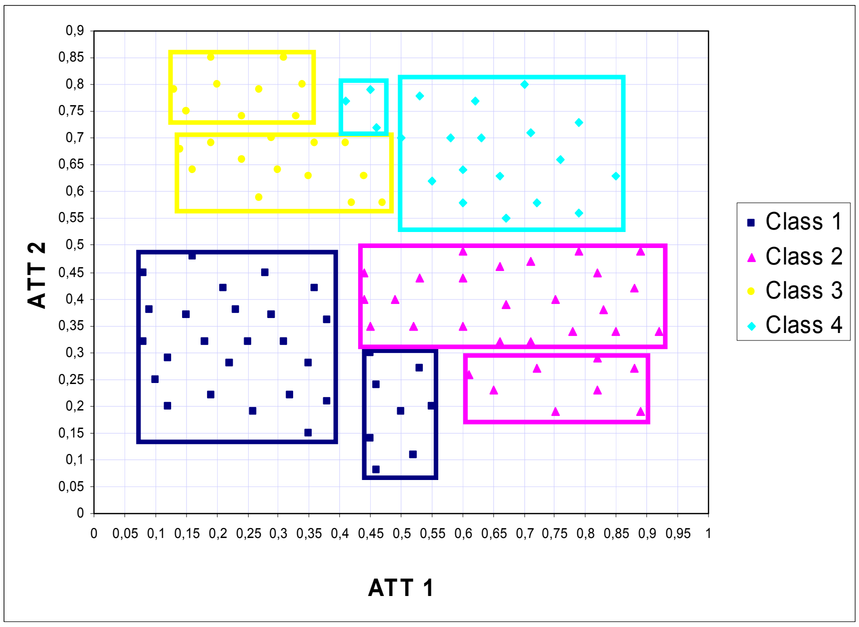

Figure 3.11 Combined hyper-boxes after the second run of IP-Box Combine model.

3.3 Testing Algorithm

3.3.1 Original Testing Algorithm

The original testing algorithm proposed in master thesis [106] is briefly explained in this section. If a new instance with an unknown class is given, it is necessary to assign this instance to one of the classes. There are two possibilities for a new instance when determining its class:

i. the new instance is within the boundaries of a hyper-box,
ii. the new instance is not enclosed in any of the hyper-boxes determined in the training problem.

When the first possibility is realized for the new instance, the classification is made by directly assigning this instance to the class that was represented by the hyper-box enclosing the data point. In the case when the second possibility applies, the assignment of the new instance to a class requires some analysis. If the instance is within the lower and upper bounds of all but not one of the attributes (i.e., *m'*) defining the box, then the shortest distance between the new instance and the hyper-box is calculated using the minimum distance between hyper-planes defining the hyper-box and the new instance. The minimum distance between the new data point *i* and the hyper-box is calculated using Eq. (3.43) considering the fact that the minimum distance is given by the normal of the hyper-plane.

$$DH_{il} = \min_{n}\{|a_{im'} - X_{lm'n}|\} \tag{3.43}$$

When the data point is between the bounds of smaller than or equal to M-2 attributes, then the smallest distance between the point and the hyper-box is obtained by calculating the minimum distance between edges of the hyper-box and the new point. An edge is a finite segment consists of the points of a line that are between two possible pairs of extreme points EP_{lj} and EP_{lt} where *j* and *t* represent the rank of extreme points. As the number of extreme points for a given box is 2^M but the number of edges is $M2^{M-1}$, not all of the indexes will be used for edge calculation. This issue will be controlled by given the possible extreme point combinations as a set, EPP (extreme point pairs). Cardinality of EPP set is $M2^{M-1}$. The value of attribute m for data point i is represented by the parameter a_{im} and ep_{ljm} and ep_{ltm} are the values of attribute m for two possible pairs of extreme points j and t. The minimum distance between the new data point i and one of the segment of the hyper-box determined by two extreme points is calculated using Eq. (3.50).

$$w_{iljtm} = a_{im} - ep_{ljm} \tag{3.44}$$

$$v_{iljtm} = ep_{ljm} - ep_{ltm} \tag{3.45}$$

$$C1_{iljtm} = \frac{\sum_m w_{iljtm} v_{iljtm}}{\sqrt{\sum_m w_{iljtm}^2}\sqrt{\sum_m v_{iljtm}^2}} \tag{3.46}$$

$$C2_{iljtm} = \frac{\sum_m v_{iljtm} v_{iljtm}}{\sqrt{\sum_m v_{iljtm}^2}\sqrt{\sum_m v_{iljtm}^2}} \tag{3.47}$$

$$b_{iljtm} = C1_{iljtm} / C2_{iljtm} \tag{3.48}$$

$$pb_{iljtm} = ep_{ljm} + b_{iljtm} v_{iljtm} \tag{3.49}$$

$$DED_{il} = \min_{\substack{j,t: \\ (j,t)\in EPP}} \left\{ \sqrt{\sum_m (a_{im} - pb_{iljtm})^2} \right\} \tag{3.50}$$

When data point is not within the lower and upper bounds of any attributes defining the box, then the shortest distance between the new point and the hyper-box is calculated using the minimum distance between extreme points of the hyper-box and the new data. The minimum distance between the new data point i and one of the extreme points ep_{ljm} of the hyper-box is calculated using Eq. (3.51).

$$DEP_{il} = \min_{j}\left\{\sqrt{\sum_{m}(a_{im} - ep_{ljm})^2}\right\} \tag{3.51}$$

The following algorithm assigns a new data point *i* with attribute values a_{im} to class *k*:

Step 0: Initialize $inAtt_{lm}$=0.

Step 1: For each *l* and *m*, if $X_{lmn} \le a_{im} \le X_{lmn'}\ \forall n = lower, n' = upper$, set $inAtt_{lm} = inAtt_{lm} + 1$.

Step 2: If $inAtt_{lm} = M$, then go to Step 3. Otherwise, continue. If $inAtt_{lm} \le M-1$, then go to Step 4.

Step 3: Assign the new data point to class *k* where ybc_{lk} is equal to 1 for the hyper-box in Step 2. Stop.

Step 4: If $inAtt_{lm} = M-1$, then $dist_{il} = DH_{il}$.
If $0 < inAtt_{lm} < M-1$, then $dist_{il} = DED_{il}$.
If $inAtt_{lm} = 0$, then $dist_{il} = DEP_{il}$.

Step 5: Select the minimum between $\min_{l}\{dist_{il}\}$ to determine the hyper-box *l* that is closest to the new data point *i*. Assign the new data point to class *k* where ybc_{lk} is equal to 1 for the hyper-box *l*. Stop.

After finding the assigned classes of test instances, we must compare the assigned and original classes in order to calculate the accuracy of the proposed model. The proportion of correctly classified instances will give the efficiency and accuracy of the algorithm.

3.3.2 Improved Testing Algorithm

The original testing algorithm is computationally intractable for high-dimensional problems due to high number of extreme point calculations. Hence, an improved testing algorithm that approximates the original algorithm is developed. The testing results for large data classification problems can be computed in a very smaller amount of time with the improved testing algorithm compared to the original algorithm. The following new algorithm assigns a new data point *i* with attribute values a_{im} to class *k*:

Step 1: For each *l* and *m*,

If $a_{im} > X_{lmn}$ where $n = upper$, then $d_{ilm} = (a_{im} - X_{lmn})^2$.

If $a_{im} < X_{lmn'}$ where $n' = lower$, then $d_{ilm} = (X_{lmn'} - a_{im})^2$.

If $X_{lmn'} \leq a_{im} \leq X_{lmn}$ where $n = upper$ and $n' = lower$, then $d_{ilm} = 0$.

Step 2: Calculate distance from data point *i* to box *l* by using Equation 3.1.

$$Ndist_{il} = \sqrt{\sum_m d_{ilm}} \tag{3.52}$$

Step 3: Select the minimum between $\min_l \{Ndist_{il}\}$ to determine the hyper-box *l* that is closest to the new data point *i*. Assign the new data point to class *k* where ybc_{lk} is equal to 1 for the hyper-box *l*. Stop.

3.3.3 Comparison of Original and Improved Testing Algorithms

There exists four possible cases for the position of an instance *i* with respect to a hyper-box *l* in the original testing algorithm (Figure 3.12). These cases can be listed as follows:

Case I: Instance *i* is enclosed by the hyper-box *l.*

Case II: Instance *i* is within the lower and upper bounds of all but not one of the attributes (*m'*) of hyper-box *l.*

Case III: Instance *i* is between the bounds of smaller than or equal to M-2 attributes of hyper-box *l.*

Case IV: Instance *i* is not within the lower and upper bounds of any attributes of hyper-box *l.*

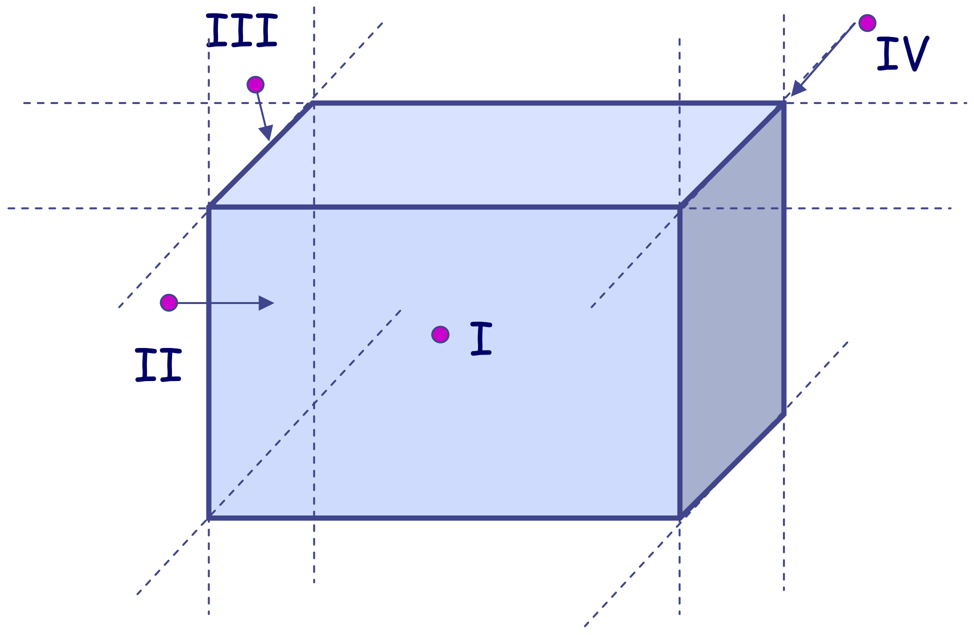

Figure 3.12 The possible positions of an instance with respect to a hyper-box.

The following analysis can be done for each case.

Case I: If instance *i* is inside the hyper-box *l*, then it is directly assigned to the corresponding class of hyper-box *l* in the original testing algorithm. Similarly, in the improved testing algorithm $X_{lmn'} \le a_{im} \le X_{lmn}$ holds and d_{ilm} will be 0 for each attribute *m*. This will result in $d_{il} = 0$. Hence, the closest hyper-box to that instance *i* will be hyper-box *l* and instance *i* will be assigned to the corresponding class of hyper-box *l*. Therefore, improved testing algorithm gives the same results as the original algorithm for Case I.

Case II: If an instance *i* is within the lower and upper bounds of all but not one of the attributes (*m'*) of hyper-box *l*, minimum distance form that instance *i* to the hyper-box *l* is calculated by using Equation 3.43 in the original testing algorithm. For the improved testing algorithm, as $X_{lmn'} \le a_{im} \le X_{lmn}$ holds for all attributes except m', d_{ilm} will be zero for those attributes and $d_{ilm'}$ will be greater than zero (Equation 3.53). Hence, distance from instance *i* to the hyper-box *l* is calculated using Equation 3.54 in the improved testing algorithm. As Eq. (3.54) and Eq. (3.43) are identical, both of the testing algorithms are identical for Case II.

$$d_{ilm'} = \min_n \left\{ (a_{im'} - X_{lm'n})^2 \right\} \tag{3.53}$$

$$d_{il} = \sqrt{d_{ilm'}} = \min_n \left| a_{im} - X_{lmn} \right| \tag{3.54}$$

Case III: If an instance *i* is between the bounds of smaller than or equal to *M*-2 attributes of hyper-box *l*, the original algorithm will calculate the distances from instance *i* to each edge of the hyper-box *l*. Then, it selects the smallest one from $m2^{m-1}$ edges as given in Eq. (3.50). On the other hand, the improved algorithm will find out the closest extreme point of the hyper-box *l* that is the one of the extreme points of the closest edge found with the original algorithm. Then, the improved algorithm calculates the Euclidean distance from instance *i* to that extreme point. Hence, the improved algorithm's distance value will always be greater than the distance value of the old algorithm.

In order to prove this more formally, assume that the closest extreme point of hyper-box *l* to instance *i* is $(X_{l1upper}, X_{l2upper}, \ldots, X_{lkupper}, \ldots, X_{lmupper})$. For the improved algorithm, distance from instance *i* to hyper-box *l* is calculated as in Eq. (3.55).

$$d_{il} = \sqrt{(a_{i1} - X_{l1upper})^2 + \ldots + (a_{ik} - X_{lkupper})^2 + \ldots (a_{im} - X_{lmupper})^2} \tag{3.55}$$

As neighboring extreme points have (m-1) attribute values in common, the closest edge to instance *i* will be the one with an end point of $(X_{l1upper}, X_{l2upper}, \ldots, X_{lkupper}, \ldots, X_{lmupper})$. Assume the other end point of this edge is $(X_{l1upper}, X_{l2upper}, \ldots, X_{lklower}, \ldots, X_{lmupper})$ as only one attribute value changes for neighbor extreme points. Then, the closest point on that edge to instance *i* is

$(X_{l1upper}, X_{l2upper}, \ldots, (X_{lklower} + b(X_{lkupper} - X_{lklower})), \ldots, X_{lmupper})$ where *b* is that ratio that shows how far instance i from start point of that edge. This *b* is given in Eq. (3.48). Hence, from Eq. (3.50), the original algorithm gives the minimum distance from instance *i* to hyper-box *l* as follows:

$$origd_{il} = \sqrt{(a_{i1} - X_{l1upper})^2 + \ldots + (a_{ik} - X_{lklower} - k(X_{lkupper} - X_{lklower}))^2 + \ldots (a_{im} - X_{lmupper})^2}$$

(3.56)

All terms of d_{il} and $origd_{il}$ are equal to each other except $(a_{ik} - X_{lkupper})^2$ and $(a_{ik} - X_{lklower} - k(X_{lkupper} - X_{lklower}))^2$. We only need to compare these terms to give the superiority relationship between d_{il} and $origd_{il}$.

Claim: $d_{il} \geq origd_{il}$.

Proof: As mentioned before, all terms are equal in these distance values except $(a_{ik} - X_{lkupper})^2$ and $(a_{ik} - X_{lklower} - k(X_{lkupper} - X_{lklower}))^2$. Hence, we need to compare these two terms in order to conclude. As closest extreme point consists of $X_{lkupper}$, then $0.5 \leq k \leq 1$ and $a_{im} > X_{lmupper}$ holds.

$$(a_{ik} - X_{lkupper})^2 \overset{?}{\geq} (a_{ik} - X_{lklower} - k(X_{lkupper} - X_{lklower})^2 \tag{3.57}$$

$$\left|a_{ik} - X_{lkupper}\right| \overset{?}{\geq} \left|a_{ik} - X_{lklower} - k(X_{lkupper} - X_{lklower})\right| \tag{3.58}$$

$$a_{ik} - X_{lkupper} \overset{?}{\geq} a_{ik} - X_{lklower} - k(X_{lkupper} - X_{lklower}) \tag{3.59}$$

$$(k-1)X_{lkupper} \overset{?}{\geq} (k-1)X_{lklower} \tag{3.50}$$

$$X_{lkupper} \overset{?}{\geq} X_{lklower} \tag{3.61}$$

Since $X_{lkupper}$ is always greater than or equal to $X_{lklower}$, the claim $d_{il} \geq origd_{il}$ is true. In the same manner, the case where instance i is closer to $X_{lklower}$ can be proved. Therefore, the improved testing algorithm gives distance values greater than or equal to the original distance value for Case III.

Case IV: If an instance *i* is not within the lower and upper bounds of any attributes of hyper-box *l*, the original algorithm calculates the distances from instance *i* to each extreme points of hyper-box *l*. Then, it will select the smallest one from 2^m extreme points as given in Eq. (3.51). On the other hand, the proposed improved algorithm tries to find the closest bound (either lower or upper) for each attribute. Then, the closest extreme point will be found out by these closest bounds. Hence, the same distance value will be obtained as in the original testing algorithm. Both algorithms give identical distance values for Case IV.

Therefore, the improved testing algorithm is an approximation of the original testing algorithm. In Cases I, II and IV, calculated distance values will be same. On the other hand, for Case III improved testing algorithm will give a higher distance value. Hence, the improved testing algorithm is an approximation of the original one.

3.3.4 Computational Complexities of the Original and Improved Testing Algorithms

The original testing algorithm has a poor computational performance on data sets with large number of attributes. The improved algorithm is an approximation of the original algorithm. Therefore, a worse performance can be expected from the new algorithm. However, the computational complexity of the improved algorithm is far superior to the original one. Therefore, we compare the computational complexities of two testing algorithms. The number of algebraic operations for the original testing algorithm is $O(M2^{M-1})$ whereas that for the new testing algorithm is $O(LM)$ (see Table 3.2). Thus, the original testing algorithm is an exponential algorithm that depends one the number of attributes *M*. However, the improved testing algorithm is a polynomial algorithm that depends on the number of hyper-boxes *L* and number of attributes *M*. Hence, the improved testing algorithm is preferable in the case of data classification problems with large number of attributes.

Table 3.2. Computational complexities of two testing algorithms.

Original Testing Algorithm		**New Testing Algorithm**	
Place	Computation Time	Place	Computation Time
Step 1	O(LM)	Step 0	O(LM)
Step 4	O(N), O(2^M), O($M2^{M-1}$)	Step 3	O(L)
Step 5	O(L)		
Overall Complexity	O($M2^{M-1}$)	***Overall Complexity***	O(LM)

3.4 Illustrative Example

We applied the proposed three-stage MILP based approach on set of 105 training data points in four different classes given in Figure 3.13.

3.4.1 Training Part

When we apply proposed three-stage algorithm, we first calculate the boundaries of classes and compare whether they overlap or not. As shown in Figure 3.14, overlapping between the classes exists. The instances that are enclosed by other classes are identified as 'problematic instances'. For this data set, there exist 18 data points which fall into the bounds of other classes. These problematic instances are enclosed by dashed points in Figure 3.15. Using these problematic instances, IP-Seed model is solved to find a seed for each class. Seeds are indicated with circles in Figure 3.15. Once four seeds to be assigned to the four classes are determined, we solve MILP model for these 'problematic instances' with fixed assignment of these seeds. The constructed hyper-boxes for these problematic instances are shown in Figure 3.16.

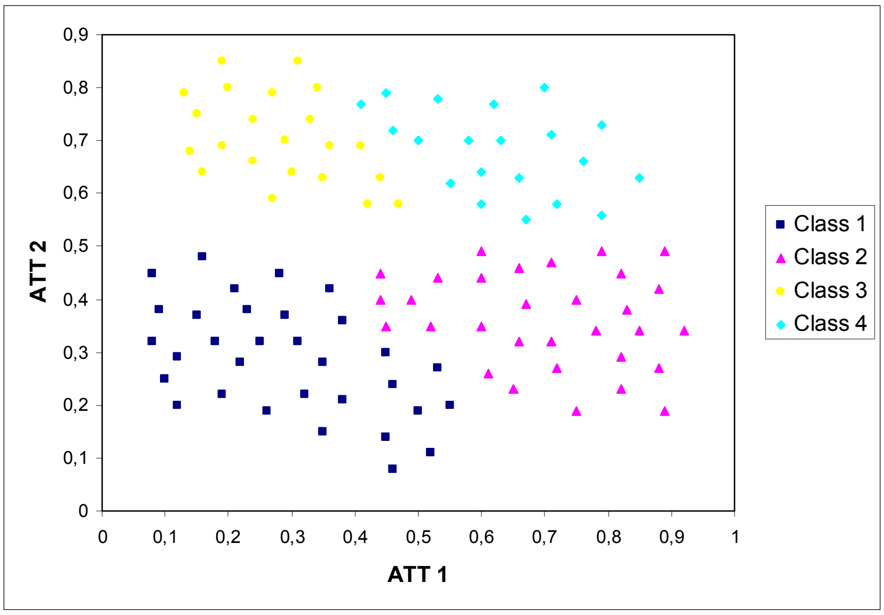

Figure 3.13 Data points in the illustrative example and their graphical representation.

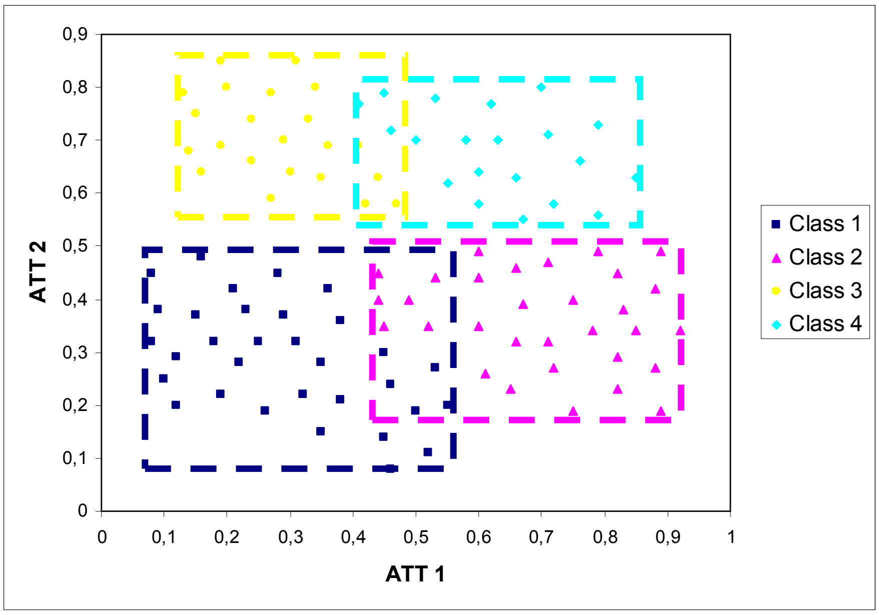

Figure 3.14 Maximum and minimum attribute values for each class.

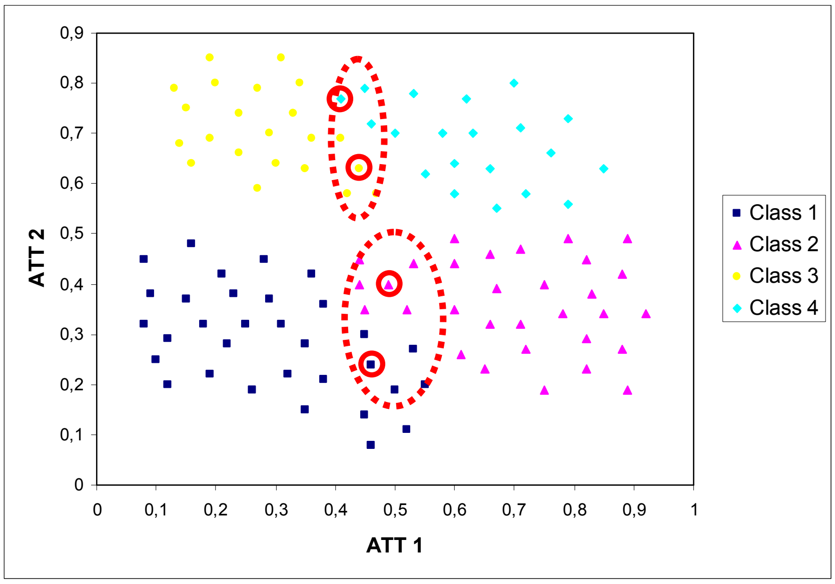

Figure 3.15 Problematic instances are enclosed by dashed points and seeds with circles.

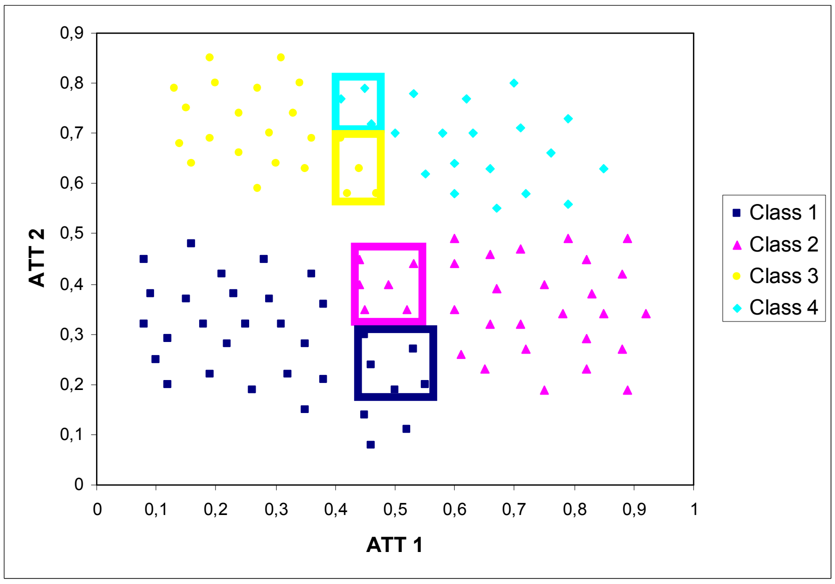

Figure 3.16 Constructed hyper-boxes for problematic instances.

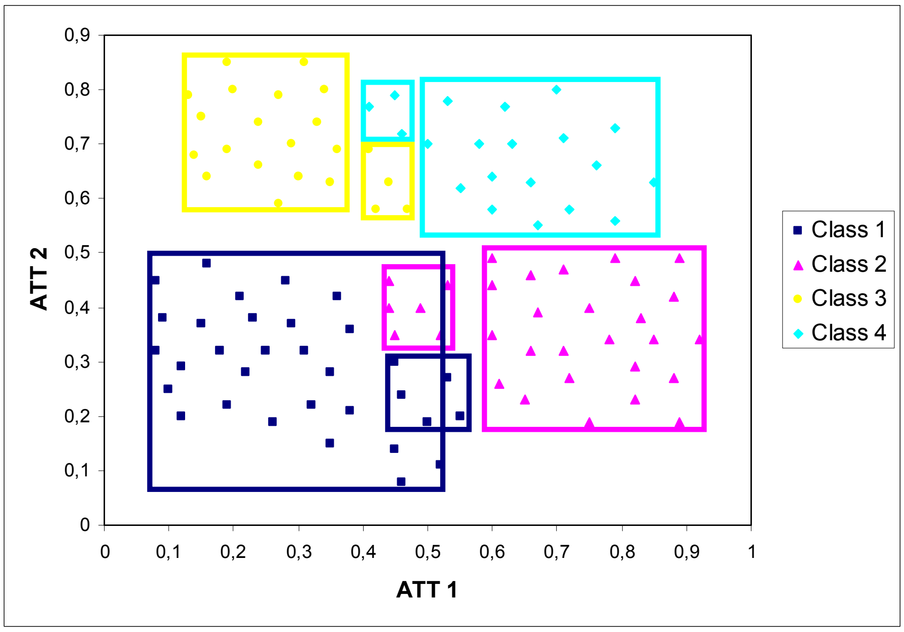

Figure 3.17 Defined and constructed hyper-boxes for illustrative example.

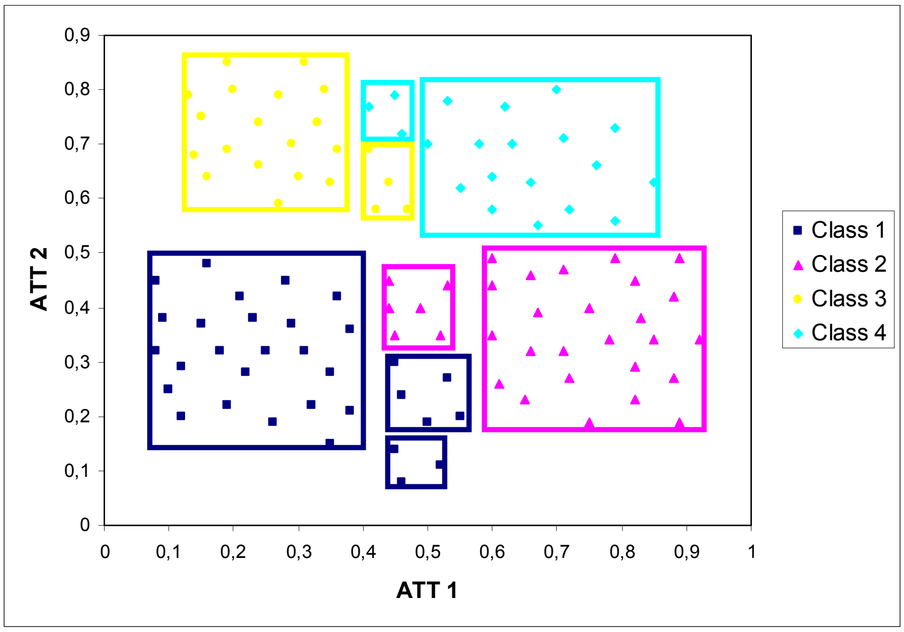

Figure 3.18 Hyper-boxes after intersection elimination for illustrative example.

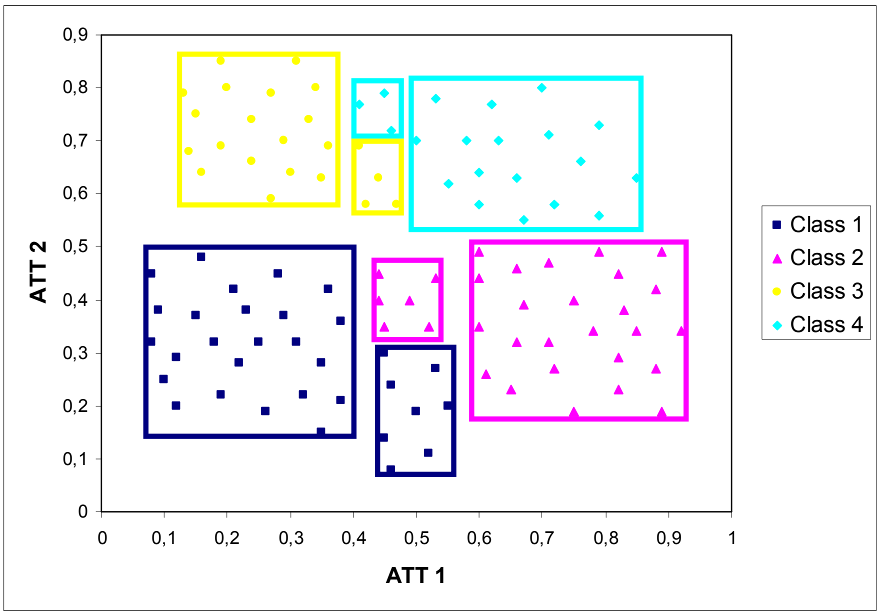

Figure 3.19 Final solution for illustrative example.

The next step is the assignment of non-problematic instances. New hyper-boxes for each class are defined and remaining 87 non-problematic instances are assigned to the hyper-boxes that correspond to their own classes. Then, the bounds of the newly defined hyper-boxes are calculated by obtaining the maximum and minimum attribute values of instances belonging to them (Figure 3.17). As it can be seen from Figure 3.17, there are some intersections between constructed and defined hyper-boxes. In order to get rid of these intersections, instances in the defined hyper-boxes are separated one by one until intersections are eliminated. Then, the eliminated instances are grouped in a new hyper-box. Resulting hyper-boxes do not intersect each other as shown in Figure 3.18. After that, IP-Box Combine model is studied and the feasible combination of hyper-boxes is obtained (Figure 3.19). The final solution for this illustrative example is found. At last, without any misclassifications of training set instances, 8 hyper-boxes are obtained.

Hence, the proposed three-stage MILP approach categorized the 105 training instances into their corresponding classes with a training accuracy value of 100%. The characteristics of each of the steps of proposed approach on illustrative example are given in Table 3.3.

Table 3.3 Problem characteristics for illustrative example.

Steps of 3-Stage Approach	**Problem Characteristics**					
	# of Nodes	**# of Iterations**	**# of Constraints**	**# of BVar**	**# of CVar**	**CPU (sec.)**
Problematic Instances	- - -	- - -	- - -	- - -	- - -	0.093
Seed Finding	0	0	59	72	0	0.078
MILP with Seeds	0	22	1509	858	265	0.265
Defined Hyper-boxes	- - -	- - -	- - -	- - -	- - -	0.063
Intersection Elimination	- - -	- - -	- - -	- - -	- - -	0.203
Box Combination	0	0	4045	222	0	0.109
Testing	- - -	- - -	- - -	- - -	- - -	0.016

3.4.2 Testing

After classifying the training data perfectly, the 52 test instances (shown in Figure 3.20) are assigned to the constructed hyper-boxes by applying the improved testing algorithm. After improved test set analysis, it is observed that all of test instances are assigned to their original classes. Hence, accuracy of the proposed three-stage approach is 100% for this illustrative example using the testing algorithm.

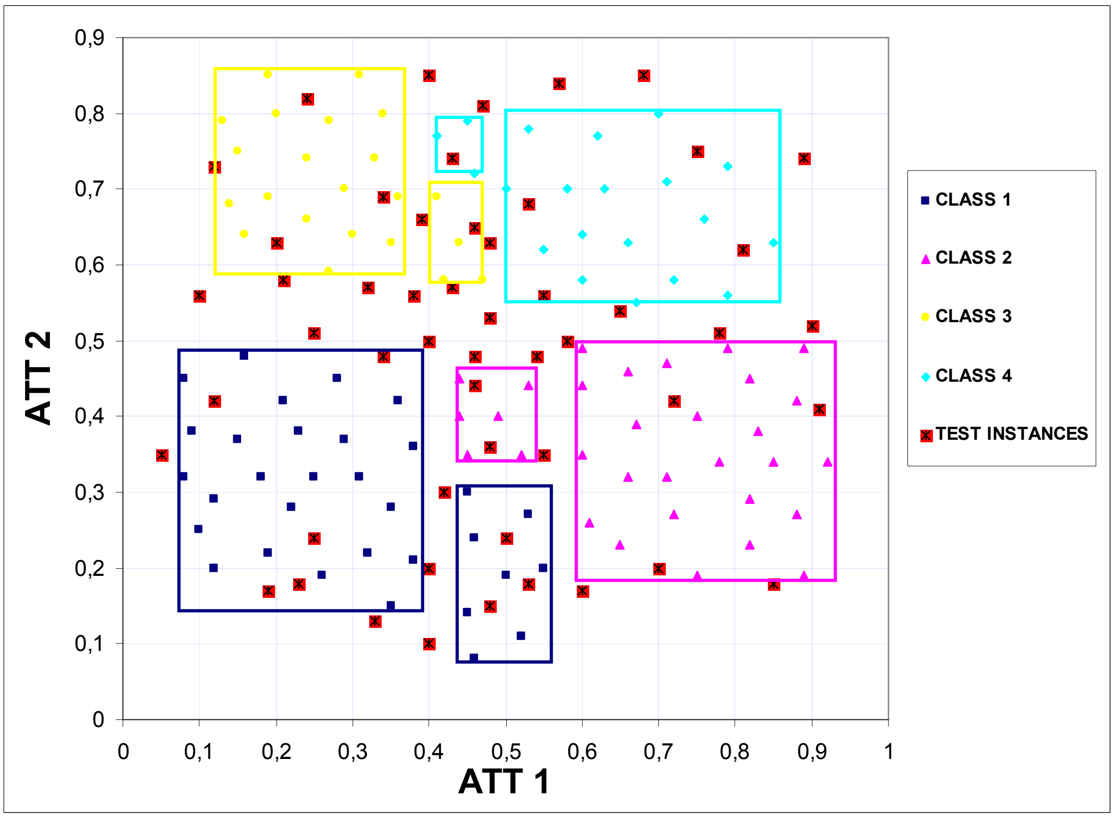

Figure 3.20 Test instances for illustrative example.

On the other hand, the same illustrative example is studied with different types of classifiers available in the well-known Weka. Weka is a collection of machine learning algorithms for data mining tasks including data classification [107]. In Table 3.4, different classification methods and their accuracy values are listed. The best accuracy value is 96.1% received by the classifier NNge (Nearest neighbor like algorithm using non-nested generalized exemplars).

Table 3.4 Accuracies of different data classification methods for illustrative example.

Classifier	Accuracy	Classifier	Accuracy
BayesNet	84.6%	Decorate	90.3%
NaiveBayes	92.3%	END	88.4%
NaiveBayesSimple	94.2%	FilteredClassifier	84.6%
NaiveBayesUpdateable	92.3%	LogitBoost	82.6%
Logistic	86.5%	MultiClassClassifier (RBF)	88.4%
MultiLayerPerceptron	88.4%	MultiClassClassifier (MultiLayerPerceptron)	86.5%
RBFNetwork	94.2%	RandomCommittee	92.3%
SimpleLogistic	88.4%	BFTree	92.3%
SMO	92.3%	J48	84.6%
IB1	92.3%	NBTree	94.2%
IB2	92.3%	RandomForest	92.3%
IB3	94.2%	RandomTree	88.4%
IB4	92.3%	REPTree	94.2%
IB5	94.2%	SimpleCart	92.3%
IB6	94.2%	NNge	**96.1%**
KStar	94.2%	Bagging	94.2%
LWL	94.2%	Ridor	92.3%
AttributeSelectedClassifier	88.4%	ClassificationviaRegression	90.3%

As a result, suggested three-stage approach performs better than other data classification methods that are listed in Table 3.4 for this illustrative example. Thus, this new method can be attractive for real life data classification problems. For further investigation to the performance of the developed MILP based algorithm, distinct benchmark problems are examined in the next chapter of the thesis.

3.4.3 The Original and New Testing Algorithms' Performances on New Thyroid Data Set

In this part of the study, the efficiencies of original and new testing algorithms are compared on new thyroid dataset [108]. This data set is composed of 215 samples with 5 different attribute values and 3 different classes: euthyroidism (class 1), hypothyroidism (class 2), or hyperthyroidism (Class 3). In this dataset, 150 of instances belong to class 1, 35 of them belong to class 2 and remaining 30 belong to class 3.

For thyroid data set, 10-fold cross-validation approach is used to estimate the performance of three-stage MILP based approach with both original and new testing algorithms.

In Table 3.5, results for the new and original testing algorithms are listed. As it is seen in Table 3.5, the new testing algorithm has better in overall accuracy for thyroid dataset. For the runs 2, 3, 5, 7 and 8, both algorithms give the same accuracy values. On the other hand, in runs 1, 4, 6 and 10 the new testing algorithm has a higher accuracy value. Interestingly, the original testing algorithm has 100% accuracy for run 9, which is more accurate than the new testing algorithm. To sum up, we could not conclude that the new testing algorithm is always better than the original algorithm with respect to accuracy. However, it gives better results on most of the cases and has higher average classification accuracy for thyroid data set.

Table 3.5 Prediction results for Thyroid data set for original and new testing algorithm.

# of run	**Accuracy with the original testing algorithm**	**Accuracy with the new testing algorithm**
1	90.90%	95.45%
2	95.45%	95.45%
3	95.45%	95.45%
4	86.36%	90.90%
5	100%	100%
6	95.23%	100%
7	100%	100%
8	95.23%	95.23%
9	100%	90.47%
10	76.19%	80.95%
Overall	**93.48%**	**94.39%**

Chapter 4

COMPUTATIONAL RESULTS ON PROTEIN FOLDING TYPE PREDICTION

The performance of proposed three-stage approach is evaluated on distinct protein folding type prediction benchmark data sets. The prediction results and comparisons with other data classification methods are examined in this chapter.

4.1 Protein Folding Type Prediction Problem

The prediction of protein folding type is a typical multi-group data classification problem. The are four different classes; all-alpha (α), all-beta (β), alpha+beta (α+β), alpha/beta (α/β). 20 amino acid compositions constitute the attributes of protein folding type prediction problem.

4.2 Protein Folding Type Data Sets

In order to observe the performance of the proposed approach, the following four data sets from [97] are tested: 138 domains in Table A.1, 253 domains in Table A.2, 359 domains in Table A.3, 1601 domains in Table A.4, 225 Domains in Table A.5, 510 Domains in Table A.6, 2438 Domains in Table A.7. Moreover, two data sets from [100] are studied: 277 Domains in Table A.8 and 498 Domains in Table A.9. Finally, two more data sets from [23] are tested: 1189 Domains in Table A.10 and 25PDB in Table A.11. Each of these data sets is constructed from SCOP [91] and Protein Data Bank [90]. The unit of classification in the SCOP database is usually the protein domain. Small proteins and most medium-size proteins have single domain. Domains in large proteins are usually classified individually. Therefore, the sequence of a domain considered here is either the whole chain or a partial chain of a protein. Each domain is represented by a symbol of X|Y, where first four character of X is the corresponding PDB code and the fifth character indicates the specific chain of the protein. If it is _, then the corresponding protein has only one chain. If Y=W.C., it means the domain is constituted by the whole chain. Otherwise, Y contains two number to indicate starting and end points along the sequence.

In the SCOP database, protein domains are classified into the following 11 categories [91]: (1) all-α proteins; (2) all-β proteins; (3) α/β proteins; (4) α+β proteins; (5) multi-domain proteins; (6) membrane and cell surface proteins; (7) small proteins; (8) coiled coils proteins; (9) low resolution proteins; (10) peptides; and (11) designed proteins. Usually, only the first four categories are considered for computational prediction purposes as they include significant majority of the protein sequences.

For 138, 253, 359, 225, 510, 277, 498, 1189 and 25PDB protein data sets, they are assumed to have four different classes. On the other hand, for 1601 and 2438 protein domains seven different structural classes, i.e. all α, all β, α+β, α/β, multi domain (μ), small protein (σ) and peptides (ρ), were used. Details related to these seven classes were given in [23, 97].

The leave-one-out (LOO) results of 138, 253, 359 and 1601 data sets are given in [97] and [109]. Moreover, the prediction quality is also examined by independent training and test data sets as in [97] and [110]. The training data set is composed of 225 protein domains and the corresponding test data set contains 510 protein domains. Furthermore, 1601 protein domains are used as training set in order to test the performance on 2438 protein domains. On the other hand, LOO results of 277 and 498 domain data sets are given in [100], [104] and [109]. Finally, 10-fold cross-validation (10FCV) results of 1189 and 25PDB data sets are mentioned in [23].

4.3 Classification Algorithms

In order to compare the results of proposed MILP approach, WEKA classification algorithms J48, RBF Network, Logistic, Naïve Bayes (NB), SMO, Random Forest (RF) and IB1 are also studied (Table 4.1). Optimized parameter values of these WEKA classifiers given by [23] are used to perform the studies on the given data sets. Moreover, well-known support vector machine implementation LibSVM given by [111] is also studied to observe the accuracy values. For each of the data sets, parameters related to SVM algorithm are optimized by performing 10FCV validation with different combinations of cost and gamma values. The optimal values that achieve the highest 10FCV accuracy are used to obtain the LOO results for each data set (Table 4.2).

Table 4.2 Optimal parameter values of LibSVM for each of the data sets.

Data Sets	**Kernel Type**	**c (Cost)**	**g (Gamma)**
138 Protein Domains	Radial Basis Function	2048	8
253 Protein Domains	Radial Basis Function	8192	8
359 Protein Domains	Radial Basis Function	512	8
277 Protein Domains	Radial Basis Function	2048	8
498 Protein Domains	Radial Basis Function	2048	8
225&510 Protein Domains	Radial Basis Function	32	2
1601&2438 Protein Domains	Radial Basis Function	128	8
1189 Protein Domains	Radial Basis Function	512	0.5
25PDB Protein Domains	Radial Basis Function	8	8

Table 4.1 Summary of the applied classification algorithms of WEKA.

Classifier	Reference	Short Description
Naïve Bayes	[112]	• Class for a Naive Bayes classifier using estimator classes. • Numeric estimator precision values are chosen based on analysis of the training data.
RBF Network	[113]	• Class that implements a normalized Gaussian radial basis function network. • It uses the k-means clustering algorithm to provide the basis functions and learns either a logistic regression (discrete class problems) or linear regression (numeric class problems) on top of that. • It standardizes all numeric attributes to zero mean and unit variance.
IB1	[114]	• IB1-type classifier. • Uses a simple distance measure to find the training instance closest to the given test instance, and predict the same class as this training instance. • If multiple instances are the same (smallest) distance to the test instance, the first one found is used.
J48	[115]	• Class for generating an unpruned or a pruned C4.5 decision tree.
Random Forest	[116]	• Decision tree type algorithm • Class for constructing random forests.
JRip	[117]	• This class implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), which is proposed by William W. Cohen as an optimized version of IREP.
SMO	[118]	• Implements John C. Platt's sequential minimal optimization algorithm for training a support vector classifier using polynomial kernels. • Transforms output of SVM into probabilities by applying a standard sigmoid function that is not fitted to the data.
Logistic	[119]	• Class for building a logistic regression model using LogitBoost. • Incorporates attribute selection by fitting simple regression functions in LogitBoost.

Furthermore, the existing results of distance-based classification methods based on Hamming Distance (HD), Euclidean Distance (ED) and Component-coupled (CC) algorithms given in [97] and [100], the reported results of SVM algorithm used in [110] and [109], and the existing result of Neural Networks method given in [104] are also investigated for comparison.

4.4 Results for Independent Data Sets

Using the 225 training set samples given in [97] (Table A.5), the proposed three-stage MILP model is solved in GAMS [120] using ILOG CPLEX Solver version 10.0 [121] on a notebook computer with Inter Pentium M 1.73 Ghz Processor and 512 MB of RAM. The characteristics of the constructed model for 225 training samples are listed in Table 4.3.

After classifying the training data perfectly (self-consistency test result is 100%), the test set given in Table A.6 is assigned to constructed hyper-boxes by applying the testing algorithm. The assignment of data in the test set to structural classes is done without a prior knowledge on their membership to a class. For each member of the test data set, testing algorithm is applied and an assignment to a structural class is done. After all, the accuracy of the developed model is checked by comparing the original and assigned structural classes of proteins. At the end of the testing, it is realized that 489 proteins in the test set are correctly classified. On the other hand, 21 proteins are misclassified.

Table 4.3 Characteristics of the MILP model for 225 training samples.

ITEM	VALUE
# of continuous variables	2401
# of binary variables	30750
# of constraints	52495
# of nodes	283
# of iterations	22945
Solver Memory (MB)	12
CPU time (sec)	29.343

Table 4.4 Performance results for the 510 protein domains in the test set.

Methods	**Class-based Accuracy**				**Overall Accuracy**
	α	**β**	**α+β**	**α/β**	
MILP	93.58%	96.15%	96.32%	97.04%	**95.88%**
IB1	90.83%	94.62%	98.53%	97.78%	95.68%
SVM	NA	NA	NA	NA	94.90%
Random Forest	93.58%	96.15%	86.76%	95.56%	92.94%
Component-coupled	74.31%	90.00%	87.50%	91.85%	86.47%
J48	80.73%	59.23%	82.35%	79.26%	75.29%
LibSVM	63.30%	78.46%	50.00%	42.22%	58.04%
RBF Network	46.79%	58.46%	51.47%	66.67%	56.27%
Logistic	68.81%	77.69%	30.15%	50.37%	55.88%
Naïve Bayes	45.87%	69.23%	24.26%	77.78%	54.50%
SMO	50.46%	49.23%	49.26%	50.37%	49.80%
JRip	18.35%	66.15%	74.26%	26.67%	47.64%
Euclidean Distance	50.46%	75.38%	23.53%	41.48%	47.25%
Hamming Distance	60.55%	73.08%	22.06%	36.30%	47.06%

The overall accuracy of the proposed model on 510 protein domains is 95.88%. The results of distance-based classification methods Hamming Distance, Euclidean Distance and Component-coupled algorithms [97] and the result of SVM algorithm [110] are listed in Table 4.4. Moreover, LibSVM and classifiers found in WEKA are also studied to observe the accuracy values. Proposed three-stage MILP approach gives the highest accuracy for this test set as shown in Table 4.4. IB1, instance-based classifier, has the closest accuracy value to MILP approach. SVM result given in [110] has a higher accuracy value compared to well-known support vector machine implementations SMO and LibSVM. As Cai *et al.* [110] did not provide individual accuracy values of classes and detailed confusion matrix; we could not compare classed-based accuracies. Hamming Distance and Euclidean Distance algorithm has the worst accuracy values for this data set.

In the same manner, 1601 domains data set (Table A.4) is studied by proposed three-stage MILP approach. After classifying the training data perfectly self-consistency test result is 100%), the test set composed of 2438 domains given in Table A.7 is assigned to constructed hyper-boxes by applying the testing problem algorithm. The accuracy of the developed model is checked by comparing the original and assigned structural classes of proteins. At the end of the testing, it is realized that 2318 proteins in the test set are correctly classified. On the other hand, 120 proteins are misclassified.

Table 4.5 Performance results for the 2438 protein domain in the test set.

Methods	**Class-based Accuracy**							**Overall**
	α	**β**	**α+β**	**α/β**	**μ**	**σ**	**ρ**	**Accuracy**
MILP	96.44%	95.74%	95.72%	97.25%	71.74%	87.34%	85.00%	**95.08%**
IB1	95.17%	94.18%	97.20%	95.48%	89.13%	94.30%	65.00%	95.03%
SVM	NA	NA	NA	NA	NA	NA	NA	94.50%
RF	93.38%	92.76%	94.74%	92.14%	86.96%	96.84%	75.00%	93.23%
J48	83.72%	87.93%	87.34%	88.41%	71.74%	85.44%	35.00%	86.54%
LibSVM	79.39%	92.90%	79.11%	87.43%	90.00%	96.20%	0.0%	84.58%
CC	68.70%	78.27%	69.74%	86.44%	76.09%	90.51%	75.00%	77.03%
JRip	64.12%	91.05%	44.57%	51.47%	13.04%	89.24%	60.00%	65.01%
RBF	65.14%	70.17%	52.63%	68.76%	41.30%	85.44%	35.00%	64.84%
SMO	58.78%	72.44%	60.36%	64.83%	0.00%	76.58%	0.00%	63.94%
Logistic	63.87%	76.28%	51.64%	60.90%	0.00%	79.11%	0.00%	63.04%
NB	58.78%	67.05%	26.64%	72.69%	28.26%	80.38%	40.00%	56.72%
ED	56.23%	57.10%	23.52%	49.51%	50.00%	77.22%	5.00%	47.74%
HD	47.08%	58.81%	10.36%	45.58%	47.83%	74.05%	0.00%	42.38%

The overall accuracy of the proposed model on 2438 protein domains is 95.08%. In Table 4.5, accuracy results given in [97] and [110] are listed. Moreover, LibSVM and the same classifiers found in WEKA are also studied. Proposed three-stage MILP approach gives the highest accuracy for this test data set than Hamming Distance, Euclidean Distance, Component-coupled and SVM methods. However, the accuracy values of MILP approach is much closer to IB1 accuracy value. SVM result compared to well-known support vector machine classifiers LibSVM and SMO.

4.5 Results for Self-consistency Tests

For self-consistency tests, data sets with 138, 253, 359, 1601, 277 and 498 protein domains given in Appendix are used. Using these data sets, the proposed three-stage MILP model is solved in GAMS [120] using ILOG CPLEX Solver version 10.0 [121] on a notebook computer with Inter Pentium M 1.73 Ghz Processor and 512 MB of RAM. For each data set, as we will perform LOO tests, training runs are carried out. Average self-consistency test results for 138, 253, 359 and 1601 data sets are given in Table 4.6. Moreover, average self-consistency test results for 277 and 498 domains are listed in Table 4.7.

Table 4.6 Self-consistency test results for 138, 253, 359 and 1601 Domains.

Methods	**138 Domains**	**253 Domains**	**359 Domains**
Hamming Distance [97]	55.8%	52.57%	55.15%
Euclidean Distance [97]	57.25%	53.36%	52.37%
Component-coupled [97]	97.83%	95.26%	94.43%
SVM [110]	**100%**	**100%**	93%
3-Stage MILP Approach	**100%**	**100%**	**100%**

Table 4.7 Self-consistency test results for 277 and 498 Domains.

Methods	**277 Domains**	**498 Domains**
Hamming Distance [100]	62.8%	65.5%
Euclidean Distance [100]	58.8%	64.3%
Component-coupled [100]	94.2%	95.8%
NN [104]	93.5%	94.6%
SVM [109]	**100%**	**100%**
3-Stage MILP Approach	**100%**	**100%**

Self-consistency test results indicate the percentage of information grasped during the training studies that captures the relationship between amino acid composition and protein folding type. As it could be observed from Table 4.6 and Table 4.7., proposed three-stage MILP approach gives highest self-consistency results for each one of the data sets. Hence, the relationship between amino acid composition and protein folding type is fully grasped by the developed approach.

4.6 Results for Leave-one-out Tests

In this part, structural classes of leaved-out proteins are predicted by the results derived using all other proteins in the training set. LOO test results for 138 protein domains are given in Table 4.8. LibSVM method has the highest LOO test result for 138 protein domains data set with accuracy of 70.29%. Proposed MILP approach has the second best LOO accuracy value, 67.39%, for 138 protein domains data set. IB1 classifier of WEKA also has a very close result to MILP approach. Detailed comparison of these methods based on hypothesis testing is given in Section 4.8.

Table 4.8 LOO test results for 138 protein domains.

Methods	**Class-based Accuracy**				**Overall**
	α	**β**	**α+β**	**α/β**	**Accuracy**
LibSVM	80.56%	75.86%	65.85%	59.40%	**70.29%**
MILP	83.33%	79.31%	63.41%	43.80%	67.39%
IB1	61.11%	79.31%	58.54%	71.90%	66.67%
Component-coupled	77.78%	55.17%	85.37%	28.12%	63.77%
SMO	63.89%	75.86%	53.66%	53.10%	60.87%
J48	63.89%	72.41%	58.54%	50.00%	60.87%
Random Forest	66.67%	65.52%	56.10%	53.10%	60.14%
RBF Network	63.89%	62.07%	56.10%	50.00%	57.97%
SVM	52.77%	75.86%	58.50%	43.75%	57.24%
Naïve Bayes	63.89%	65.52%	34.15%	56.30%	53.62%
Logistic	61.11%	65.52%	46.34%	40.60%	52.90%
Hamming Distance	61.11%	55.17%	36.59%	43.75%	48.55%
Euclidean Distance	61.11%	51.72%	34.15%	40.62%	46.38%
JRip	50.00%	58.62%	48.78%	18.80%	44.20%

Existing and calculated LOO test results for 253 protein domains are given in Table 4.9. Proposed three-stage MILP approach has the highest LOO test result for 253 protein domains with accuracy of 87.65%. Instance-based classifier IB1 has the second best result with accuracy value of 86.45%. Random Forest classifier and LibSVM have also high classification accuracy values with respect to other methods.

Table 4.9 LOO test results for 253 protein domains.

Methods	Class-based Accuracy				Overall
	α	β	α+β	α/β	Accuracy
MILP	91.94%	85.96%	92.96%	78.69%	**87.65%**
IB1	90.32%	85.96%	80.28%	90.16%	86.45%
Random Forest	87.10%	80.70%	80.28%	83.61%	82.86%
LibSVM	88.71%	77.19%	76.06%	85.25%	81.67%
J48	80.65%	68.42%	67.61%	73.77%	72.51%
Component-coupled	84.13%	79.31%	70.49%	81.69%	63.77%
JRip	66.13%	61.40%	63.38%	55.74%	61.75%
SMO	67.74%	70.18%	52.11%	52.46%	60.15%
RBF Network	66.13%	66.67%	53.52%	52.46%	59.36%
Naïve Bayes	69.35%	59.65%	40.85%	68.85%	58.96%
SVM	84.12%	79.31%	81.96%	87.32%	57.24%
Logistic	61.29%	63.16%	49.30%	37.70%	52.58%
Hamming Distance	60.32%	60.34%	47.54%	29.58%	48.55%
Euclidean Distance	58.73%	62.07%	47.54%	35.21%	46.38%

Table 4.10 shows the LOO test results for 359 protein domains. Proposed three-stage MILP based approach has the highest LOO test result for 359 protein domains with accuracy of 96.38%. The accuracy value of the SVM method given in [110] is the second best result. However, the well-known support vector machine classifiers LibSVM and SMO have surprisingly lower results than this SVM result. Instance-based classifier IB1 and LibSVM has also higher classification accuracy values than other existing methods.

Table 4.10 LOO test results for 359 protein domains.

Methods	**Class-based Accuracy**				**Overall**
	α	**β**	**α+β**	**α/β**	**Accuracy**
MILP	98.78%	97.65%	92.47%	96.97%	**96.38%**
SVM	92.68%	96.47%	96.77%	94.94%	95.26%
IB1	93.90%	94.12%	88.17%	97.98%	93.59%
LibSVM	92.68%	90.59%	86.02%	96.97%	91.64%
Random Forest	89.02%	88.24%	82.80%	94.95%	88.85%
Component-coupled	89.02%	83.53%	78.49%	85.85%	84.12%
J48	76.83%	88.24%	69.89%	85.86%	80.22%
JRip	76.83%	74.12%	63.44%	77.78%	72.98%
RBF Network	67.07%	65.88%	53.76%	69.70%	64.06%
SMO	65.85%	69.41%	45.16%	70.71%	62.67%
Naïve Bayes	68.29%	67.06%	36.56%	73.74%	61.28%
Logistic	57.32%	65.88%	47.31%	53.54%	55.71%
Hamming Distance	57.32%	60.00%	33.33%	59.60%	52.37%
Euclidean Distance	62.20%	60.00%	34.41%	43.43%	41.22%

LOO test results for 277 protein domains are given in Table 4.11. LibSVM method has the highest LOO test result for 277 protein domains with accuracy value of 84.48%. Ib1 has a very close accuracy value of 84.11% for 277 protein data set. Proposed three-stage MILP based approach has the third highest LOO test result for 277 protein domains data set with accuracy value of 81.50%.

Table 4.11 LOO test results for 277 protein domains.

Methods	**Class-based Accuracy**				**Overall**
	α	**β**	**α+β**	**α/β**	**Accuracy**
LibSVM	82.86%	88.52%	75.83%	90.12%	**84.48%**
IB1	80.00%	88.52%	73.85%	92.59%	84.11%
MILP	87.14%	75.41%	72.31%	88.89%	81.50%
SVM	74.30%	82.00%	72.30%	87.70%	79.40%
Component-coupled	84.30%	82.00%	67.70%	81.50%	79.10%
Random Forest	75.71%	83.61%	70.77%	85.19%	79.06%
J48	77.14%	77.05%	64.62%	85.19%	76.53%
Neural Network	68.60%	85.20%	56.90%	86.40%	74.70%
RBF Network	77.14%	68.85%	53.85%	77.78%	70.03%
SMO	72.86%	75.41%	44.62%	77.78%	68.23%
JRip	64.29%	75.41%	55.38%	76.54%	68.23%
Naïve Bayes	74.29%	57.38%	47.69%	77.78%	65.34%
Logistic	71.43%	67.21%	44.62%	58.02%	60.28%
City-block Distance	72.90%	62.30%	43.10%	60.50%	59.90%
Euclidean Distance	71.40%	54.10%	41.50%	53.10%	55.20%

Table 4.12 shows the LOO test results for 498 protein domains. The overall accuracy of the proposed MILP model on 498 protein domains is 92.97%. On the other hand, the best accuracy value is 93.20% received by SVM given in [110]. However, the accuracy value of MILP approach is closer to SVM accuracy value. Moreover, the accuracy values of LibSVM and SMO classifiers are 92.17% and 76.30%, respectively, which are lower with respect to SVM result given in [110]. As they did not give any

detailed information related to predicted results for 498 data sets, we could not investigate the results in deeper.

Table 4.12 LOO test results for 498 protein domains.

Methods	**Class-based Accuracy**				**Overall**
	α	**β**	**α+β**	**α/β**	**Accuracy**
SVM	88.80%	95.20%	91.50%	96.30%	**93.20%**
MILP	91.59%	94.44%	93.80%	91.91%	92.97%
IB1	89.72%	96.83%	88.37%	95.59%	92.77%
LibSVM	91.59%	94.44%	89.92%	92.65%	92.17%
Random Forest	89.72%	92.86%	89.92%	94.12%	91.76%
Component-coupled	93.50%	88.90%	84.50%	90.40%	89.20%
Neural Network	86.00%	96.00%	86.00%	88.20%	89.20%
JRip	87.85%	88.89%	83.72%	88.24%	87.14%
J48	84.11%	88.89%	86.82%	87.50%	86.94%
SMO	71.03%	71.43%	74.42%	86.76%	76.30%
Logistic	68.22%	79.70%	65.89%	82.35%	74.29%
RBF Network	68.22%	75.40%	68.22%	74.26%	71.68%
Naïve Bayes	76.64%	72.22%	55.81%	75.00%	69.67%
Euclidean Distance	73.80%	65.10%	56.60%	60.30%	63.50%
City-block Distance	64.50%	68.30%	50.40%	67.70%	62.70%

4.7 Results for 10-Fold Cross-validation Tests

For the 1189 and 25PDB data sets, there exists 10-fold cross validation results in literature. Therefore, we investigate the performance of these data sets by applying 10-fold cross-validation (10FCV). The 10FCV test results for 1189 protein domains are given in

Table 4.13. The overall accuracy of the proposed model on 1189 protein domains is 53.30% with the highest accuracy value. LibSVM and Logistic classifiers has second and third best results for 1189 data set. On the other hand, the IB1 classifier which gives generally better results for the above data sets has the worst accuracy value for 1189 data set. This is a surprising result.

Table 4.13 10FCV test results for 1189 protein domains.

Methods	**Class-based Accuracy**				**Overall**
	α	**β**	**α+β**	**α/β**	**Accuracy**
MILP	76.23%	59.86%	36.52%	44.31%	**53.30%**
LibSVM	47.09%	65.99%	12.03%	74.55%	52.84%
Logistic	51.57%	67.35%	15.35%	66.17%	52.29%
SMO	46.19%	63.61%	8.29%	75.15%	51.37%
RBF Network	45.74%	53.40%	24.07%	71.86%	51.01%
Naïve Bayes	45.74%	50.68%	14.11%	79.04%	50.27%
Random Forest	47.53%	58.50%	21.99%	48.50%	45.15%
JRip	25.56%	45.24%	1.66%	82.63%	43.04%
J48	41.26%	48.30%	24.48%	51.20%	42.49%
IB1	39.46%	46.60%	19.08%	54.79%	41.57%

10FCV test results for 25PDB protein domains are given in Table 4.14. The overall accuracy of the proposed model on 1189 protein domains is 51.82%. The highest accuracy value is achieved by LibSVM method with 52.54%. SMO classifier has a very close accuracy value to LibSVM. MILP approach has the third best accuracy value as Logistic classifier. On the other hand, the IB1 classifier which gives generally better results for the above data sets has the second worst accuracy value for 25PDB data set.

Table 4.14 10FCV test results for 25PDB protein domains.

Methods	**Class-based Accuracy**				**Overall**
	α	**β**	**α+β**	**α/β**	**Accuracy**
LibSVM	65.69%	59.37%	29.48%	56.36%	**52.54%**
SMO	67.49%	63.66%	34.01%	40.17%	52.00%
MILP	60.95%	56.43%	53.47%	36.73%	51.82%
Logistic	66.82%	62.75%	34.24%	41.04%	51.82%
RBF Network	57.11%	52.37%	29.93%	60.69%	49.43%
Naïve Bayes	51.02%	45.82%	29.25%	69.36%	47.69%
Random Forest	58.24%	52.60%	27.44%	36.42%	44.11%
J48	49.21%	42.44%	31.29%	38.15%	40.40%
IB1	40.18%	35.89%	27.44%	49.13%	37.53%
JRip	42.89%	39.50%	2.49%	19.94%	26.59%

4.8 Statistical Analysis of the Results

In order to analyze the results in detail, sensitivity (SEN), specificity (SPE), MCC and S values of each of the protein data sets are calculated and examined (Table 4.15 - Table 4.24). The specificity values are always significantly greater compared to sensitivity. High average specificity means that the number of under predicted proteins is low. Thus, low accuracy is a result of relatively low sensitivity values. Moreover, as sensitivity values increases, the difference between sensitivity and specificity decreases. Therefore, observing high specificity values do not mean that the values of classification accuracy are good as expected.

MCC value gives the strength of relationship between the actual and predicted values. A perfect fit will give a MCC value of 1. Due to the low sensitivity for 138 Domains data set, MCC and S values are low for each of the classes. This means that the classifier could not effectively capture the characteristics of that class. For a perfect prediction, S value should be equal to 1 and 0 for vice versa. On the other hand, when we observe the results of each data set in overall, each of the classes have higher and lower MCC and S values with respect to the remaining classes. Hence, we could not say that MILP based hyper-box enclosure approach performs rather purely for any of the classes. Depending on the data sets, proposed data classification approach works well for each of the classes.

Table 4.15 Values of performance measures for the 138 protein domains.

Classifier	**SEN**	**SPE**	**MCC**				**S**			
			α	**β**	**α+β**	**α/β**	**α**	**β**	**α+β**	**α/β**
LibSVM	70.29%	89.79%	0.62	0.66	0.53	0.51	0.51	0.5	0.47	0.45
MILP	67.39%	88.59%	0.65	0.58	0.49	0.38	0.5	0.46	0.44	0.36
IB1	66.67%	89.35%	0.54	0.66	0.55	0.42	0.44	0.48	0.45	0.39
Component-coupled	63.77%	NA	NA	NA	NA	NA	NA	NA	NA	NA
SMO	60.87%	86.55%	0.48	0.64	0.34	0.31	0.4	0.44	0.33	0.32
J48	60.87%	86.38%	0.45	0.56	0.35	0.38	0.39	0.42	0.34	0.35
Random Forest	60.14%	86.01%	0.41	0.54	0.34	0.42	0.37	0.41	0.33	0.37
RBF Network	57.97%	85.54%	0.47	0.44	0.31	0.33	0.38	0.37	0.31	0.32
SVM	57.24%	NA	NA	NA	NA	NA	NA	NA	NA	NA
Naïve Bayes	53.62%	85.06%	0.44	0.41	0.23	0.24	0.35	0.34	0.26	0.26
Logistic	52.90%	84.21%	0.38	0.42	0.24	0.17	0.33	0.34	0.27	0.24
Hamming Distance	48.55%	NA	NA	NA	NA	NA	NA	NA	NA	NA
Euclidean Distance	46.38%	NA	NA	NA	NA	NA	NA	NA	NA	NA
JRip	44.20%	79.72%	0.22	0.41	0.03	-0.03	0.25	0.3	0.12	0.16

Table 4.16 Values of performance measures for the 253 protein domains.

Classifier	**SEN**	**SPE**	**MCC**				**S**			
			α	**β**	**α+β**	**α/β**	**α**	**β**	**α+β**	**α/β**
MILP	87.65%	95.87%	0.84	0.84	0.88	0.76	0.74	0.72	0.76	0.7
IB1	86.45%	95.47%	0.87	0.84	0.78	0.79	0.73	0.71	0.71	0.71
Random Forest	82.86%	94.11%	0.79	0.78	0.71	0.78	0.67	0.65	0.65	0.66
LibSVM	81.67%	93.86%	0.79	0.71	0.7	0.8	0.65	0.61	0.63	0.65
J48	72.51%	90.71%	0.7	0.6	0.55	0.61	0.54	0.5	0.49	0.51
Component-coupled	63.77%	NA	NA	NA	NA	NA	NA	NA	NA	NA
JRip	61.75%	86.54%	0.59	0.54	0.33	0.37	0.44	0.42	0.33	0.36
SMO	60.15%	86.54%	0.53	0.58	0.34	0.27	0.41	0.42	0.33	0.29
RBF Network	59.36%	86.08%	0.49	0.54	0.3	0.33	0.4	0.41	0.31	0.33
Naïve Bayes	58.96%	86.40%	0.56	0.46	0.3	0.35	0.41	0.38	0.3	0.33
SVM	57.24%	NA	NA	NA	NA	NA	NA	NA	NA	NA
Logistic	52.58%	84.07%	0.38	0.43	0.26	0.12	0.33	0.35	0.28	0.21
Hamming Distance	48.55%	NA	NA	NA	NA	NA	NA	NA	NA	NA
Euclidean Distance	46.38%	NA	NA	NA	NA	NA	NA	NA	NA	NA

For 2438 Domain data set, there are 7 different classes. Similar to above observations, specificity values are higher than the sensitivity values (Table 4.21 & Table 4.22). Furthermore, MCC and S values of classes α, β, α+β and α/β are greater than the MCC and S values of classes μ, σ and ρ. As the number of proteins belongs to the classes α, β, α+β and α/β are higher, proposed approach grasped the characteristics of these classes well. On the other hand, instances in μ, σ and ρ classes are very low with respect to the other classes. Hence, MCC and S values of these classes are low.

Table 4.17 Values of performance measures for the 359 protein domains.

Classifier	**SEN**	**SPE**	**MCC**				**S**			
			α	**β**	**α+β**	**α/β**	**α**	**β**	**α+β**	**α/β**
MILP	96.38%	98.77%	0.97	0.97	0.92	0.95	0.91	0.91	0.91	0.91
SVM	95.26%	NA	NA	NA	NA	NA	NA	NA	NA	NA
IB1	93.59%	97.81%	0.93	0.94	0.84	0.95	0.84	0.85	0.83	0.86
LibSVM	91.64%	97.20%	0.88	0.9	0.84	0.92	0.8	0.8	0.8	0.83
Random Forest	88.85%	96.29%	0.82	0.87	0.81	0.89	0.74	0.75	0.74	0.78
Component-coupled	84.12%	NA	NA	NA	NA	NA	NA	NA	NA	NA
J48	80.22%	93.31%	0.73	0.78	0.65	0.75	0.61	0.63	0.58	0.63
JRip	72.98%	90.77%	0.69	0.64	0.55	0.61	0.54	0.52	0.48	0.52
RBF Network	64.06%	87.70%	0.58	0.52	0.38	0.47	0.44	0.42	0.36	0.41
SMO	62.67%	87.35%	0.6	0.54	0.16	0.62	0.44	0.43	0.18	0.46
Naïve Bayes	61.28%	86.55%	0.58	0.52	0.27	0.41	0.43	0.41	0.29	0.37
Logistic	55.71%	84.87%	0.4	0.47	0.3	0.25	0.35	0.37	0.31	0.27
Hamming Distance	52.37%	NA	NA	NA	NA	NA	NA	NA	NA	NA
Euclidean Distance	41.22%	NA	NA	NA	NA	NA	NA	NA	NA	NA

Table 4.18 Values of performance measures for the 277 protein domains.

Classifier	**SEN**	**SPE**	**MCC**				**S**			
			α	**β**	**α+β**	**α/β**	**α**	**β**	**α+β**	**α/β**
LibSVM	84.48%	94.72%	0.76	0.82	0.75	0.82	0.67	0.68	0.65	0.71
IB1	84.11%	94.19%	0.8	0.82	0.79	0.75	0.67	0.68	0.66	0.68
MILP	81.50%	93.74%	0.7	0.73	0.73	0.82	0.63	0.61	0.61	0.67
SVM	79.40%	NA	NA	NA	NA	NA	NA	NA	NA	NA
Component-coupled	79.10%	NA	NA	NA	NA	NA	NA	NA	NA	NA
Random Forest	79.06%	92.91%	0.69	0.73	0.68	0.74	0.59	0.6	0.57	0.62
J48	76.53%	92.18%	0.66	0.69	0.56	0.76	0.56	0.56	0.5	0.61
NN	74.70%	NA	NA	NA	NA	NA	NA	NA	NA	NA
RBF Network	70.03%	89.54%	0.64	0.6	0.48	0.57	0.51	0.48	0.43	0.49
SMO	68.23%	88.74%	0.68	0.61	0.41	0.51	0.5	0.48	0.38	0.45
JRip	68.23%	88.95%	0.52	0.63	0.49	0.55	0.45	0.49	0.42	0.47
Naïve Bayes	65.34%	87.62%	0.63	0.51	0.39	0.48	0.47	0.42	0.37	0.42
Logistic	60.28%	86.31%	0.55	0.51	0.3	0.34	0.42	0.4	0.31	0.33
Hamming Distance	59.90%	NA	NA	NA	NA	NA	NA	NA	NA	NA
Euclidean Distance	55.20%	NA	NA	NA	NA	NA	NA	NA	NA	NA

Table 4.19 Values of performance measures for the 498 protein domains.

Classifier	**SEN**	**SPE**	**MCC**				**S**			
			α	**β**	**α+β**	**α/β**	**α**	**β**	**α+β**	**α/β**
SVM	93.20%	NA	NA	NA	NA	NA	NA	NA	NA	NA
MILP	92.97%	97.65%	0.89	0.93	0.9	0.9	0.82	0.84	0.84	0.84
IB1	92.77%	97.53%	0.89	0.93	0.89	0.9	0.81	0.84	0.83	0.84
LibSVM	92.17%	97.40%	0.88	0.92	0.88	0.89	0.8	0.83	0.82	0.83
Random Forest	91.76%	97.25%	0.87	0.9	0.87	0.91	0.79	0.81	0.81	0.82
Component-coupled	89.20%	NA	NA	NA	NA	NA	NA	NA	NA	NA
NN	89.20%	NA	NA	NA	NA	NA	NA	NA	NA	NA
JRip	87.14%	95.63%	0.84	0.83	0.82	0.81	0.72	0.73	0.72	0.73
J48	86.94%	95.62%	0.79	0.87	0.81	0.81	0.7	0.73	0.72	0.73
SMO	76.30%	91.84%	0.65	0.68	0.72	0.65	0.54	0.56	0.57	0.56
Logistic	74.29%	91.25%	0.6	0.73	0.59	0.64	0.51	0.56	0.51	0.54
RBF Network	71.68%	90.30%	0.62	0.62	0.6	0.56	0.5	0.51	0.5	0.49
Naïve Bayes	69.67%	89.74%	0.61	0.65	0.54	0.52	0.49	0.5	0.45	0.46
Euclidean Distance	63.50%	NA	NA	NA	NA	NA	NA	NA	NA	NA
Hamming Distance	62.70%	NA	NA	NA	NA	NA	NA	NA	NA	NA

Table 4.20 Values of performance measures for the 510 protein domains.

Classifier	SEN	SPE	MCC				S			
			α	β	α+β	α/β	α	β	α+β	α/β
MILP	95.88%	98.60%	0.93	0.94	0.95	0.95	0.89	0.9	0.9	0.9
IB1	95.68%	98.52%	0.92	0.93	0.96	0.95	0.88	0.89	0.9	0.9
SVM	94.90%	NA	NA	NA	NA	NA	NA	NA	NA	NA
Random Forest	92.94%	97.69%	0.89	0.93	0.87	0.92	0.82	0.84	0.83	0.84
Component-coupled	86.47%	NA	NA	NA	NA	NA	NA	NA	NA	NA
J48	75.29%	91.97%	0.63	0.6	0.81	0.61	0.53	0.51	0.6	0.54
LibSVM	58.04%	86.02%	0.4	0.61	0.27	0.28	0.36	0.43	0.29	0.3
RBF Network	56.27%	84.93%	0.4	0.33	0.25	0.46	0.35	0.32	0.28	0.37
Logistic	55.88%	83.60%	0.39	0.54	0.07	0.34	0.34	0.39	0.15	0.32
Naïve Bayes	54.50%	84.26%	0.42	0.39	0.1	0.42	0.34	0.34	0.2	0.35
SMO	49.80%	82.24%	0.48	0.41	0.07	0.2	0.34	0.32	0.13	0.25
JRip	47.64%	81.27%	0.32	0.4	0.18	0.14	0.27	0.32	0.18	0.22
Euclidean Distance	47.25%	NA	NA	NA	NA	NA	NA	NA	NA	NA
Hamming Distance	47.06%	NA	NA	NA	NA	NA	NA	NA	NA	NA

Table 4.21 Values of performance measures-1 for the 2438 protein domains.

Classifier	**ACC (SEN)**	**SPE**	**MCC**						
			α	**β**	**α+β**	**α/β**	**ρ**	**σ**	**μ**
MILP	95.08%	98.51%	0.91	0.94	0.94	0.96	0.9	0.93	0.82
IB1	95.03%	98.78%	0.94	0.95	0.96	0.9	0.78	0.95	0.85
SVM	94.50%	NA	NA	NA	NA	NA	NA	NA	NA
Random Forest	93.23%	98.11%	0.92	0.91	0.93	0.88	0.79	0.95	0.93
J48	86.54%	96.27%	0.79	0.84	0.83	0.82	0.57	0.9	0.8
LibSVM	84.58%	95.46%	0.78	0.86	0.77	0.75	0.86	0.97	0
Component-coupled	77.03%	NA	NA	NA	NA	NA	NA	NA	NA
JRip	65.01%	87.34%	0.66	0.46	0.56	0.53	0.58	0.86	0.29
RBF Network	64.84%	90.49%	0.58	0.57	0.44	0.46	0.36	0.79	0.32
SMO	63.94%	88.99%	0.51	0.52	0.47	0.45	0	0.79	0
Logistic	63.04%	89.22%	0.47	0.55	0.42	0.45	0	0.75	0
Naïve Bayes	56.72%	88.79%	0.52	0.49	0.26	0.35	0.33	0.77	0.16
Euclidean Distance	47.74%	NA	NA	NA	NA	NA	NA	NA	NA
Hamming Distance	42.38%	NA	NA	NA	NA	NA	NA	NA	NA

Table 4.22 Values of performance measures-2 for the 2438 protein domains.

Classifier	**ACC (SEN)**	**SPE**	**S**						
			α	**β**	**α+β**	**α/β**	**ρ**	**σ**	**μ**
MILP	95.08%	98.51%	0.85	0.89	0.88	0.87	0.54	0.74	0.57
IB1	95.03%	98.78%	0.85	0.89	0.88	0.86	0.52	0.75	0.6
SVM	94.50%	NA	NA	NA	NA	NA	NA	NA	NA
Random Forest	93.23%	98.11%	0.8	0.85	0.84	0.82	0.51	0.71	0.57
J48	86.54%	96.27%	0.66	0.73	0.72	0.7	0.47	0.59	0.5
LibSVM	84.58%	95.46%	0.63	0.72	0.67	0.66	0.48	0.59	0.43
Component-coupled	77.03%	NA	NA	NA	NA	NA	NA	NA	NA
JRip	65.01%	87.34%	0.45	0.39	0.42	0.42	0.4	0.45	0.39
RBF Network	64.84%	90.49%	0.43	0.46	0.39	0.4	0.39	0.44	0.38
SMO	63.94%	88.99%	0.41	0.43	0.41	0.39	0.38	0.43	0.38
Logistic	63.04%	89.22%	0.39	0.44	0.38	0.39	0.38	0.43	0.37
Naïve Bayes	56.72%	88.79%	0.38	0.39	0.28	0.32	0.36	0.4	0.34
Euclidean Distance	47.74%	NA	NA	NA	NA	NA	NA	NA	NA
Hamming Distance	42.38%	NA	NA	NA	NA	NA	NA	NA	NA

Table 4.23 Values of performance measures for the 1189 protein domains.

Classifier	**ACC (SEN)**	**SPE**	**MCC**				**S**			
			α	**β**	**α+β**	**α/β**	**α**	**β**	**α+β**	**α/β**
MILP	53.30%	84,91%	0.37	0.36	0.285	0.28	0.31	0.33	0.3	0.28
LibSVM	52.84%	81,90%	0.37	0.39	0.025	0.3	0.33	0.33	0.2	0.28
Logistic	52.28%	82,45%	0.36	0.39	0.045	0.27	0.32	0.33	0.21	0.27
SMO	51.37%	81,30%	0.35	0.35	0.028	0.27	0.31	0.32	0.21	0.26
RBF Network	51.00%	82,29%	0.31	0.31	0.07	0.31	0.3	0.3	0.21	0.29
Naïve Bayes	50.27%	81,26%	0.33	0.3	0.019	0.3	0.3	0.29	0.2	0.27
Random Forest	45.14%	80,85%	0.24	0.23	-0.01	0.14	0.26	0.25	0.17	0.21
JRip	43.04%	76,81%	0.18	0.23	-0.08	0.14	0.24	0.25	0.19	0.14
J48	42.49%	80,22%	0.14	0.13	-0.02	0.14	0.22	0.21	0.17	0.21
IB1	41.57%	79,49%	0.1	0.22	-0.07	0.07	0.21	0.24	0.15	0.16

Table 4.24 Values of performance measures for the 25PDB protein domains.

Classifier	**ACC (SEN)**	**SPE**	**MCC**				**S**			
			α	**β**	**α+β**	**α/β**	**α**	**β**	**α+β**	**α/β**
LibSVM	52.54%	84.20%	0.42	0.35	0.082	0.32	0.34	0.32	0.19	0.31
SMO	52.00%	83.64%	0.42	0.35	0.096	0.25	0.34	0.31	0.19	0.28
MILP	51.82%	83.74%	0.39	0.29	0.322	0.13	0.33	0.29	0.31	0.21
Logistic	51.82%	83.54%	0.4	0.33	0.107	0.26	0.34	0.31	0.2	0.29
RBF Network	49.43%	83.30%	0.36	0.29	0.038	0.3	0.31	0.29	0.16	0.29
Naïve Bayes	47.69%	83.31%	0.32	0.26	0.054	0.27	0.29	0.27	0.18	0.26
Random Forest	44.11%	80.87%	0.22	0.18	-0	0.16	0.24	0.23	0.15	0.24
J48	40.40%	79.75%	0.15	0.07	-0.04	0.12	0.21	0.18	0.12	0.21
IB1	37.53%	79.79%	0.11	0.07	-0.08	0.04	0.2	0.18	0.11	0.16
JRip	26.59%	74.05%	-0.26	-0.3	-0.1	0.03	-0	-0.01	0.14	0.16

In order to evaluate if there is any statistical significant difference between the existing and proposed data classification approaches tested on the same data sets, *P*-value (paired test) analysis are carried out. The results of *P*-value test results are given in Table 4.25 and in Table 4.26.

Table 4.25 The results of P-value analyses.

Compared Methods	**138**	**253**	**359**	**277**	**498**	**510**	**2438**	**1189**	**25PDB**
	P Value	**P Value**	**P Value**	**P Value**	**P Value**	**P Value**	**P Value**	**P Value**	**P Value**
MILP vs HD	2.91	9.44	13.5	5.58	10.77	16.15	39.23	NA	NA
MILP vs ED	3.27	9.89	15.95	6.65	10.53	16.1	36.09	NA	NA
MILP vs CC	0.37	6.29	5.53	0.71	1.17	3.63	17.57	NA	NA
MILP vs SVM	1.48	1.39	0.74	0.62	1.06	1.06	0.12	NA	NA
MILP vs LibSVM	0.52	1.87	2.68	0.93	0.48	14.35	12.13	0.22	0.42
MILP vs SMO	1.13	7.04	11.19	4.09	7.29	16.54	26.94	0.9	0.1
MILP vs NN	NA	NA	NA	1.93	1.17	NA	NA	NA	NA
MILP vs IB1	0.13	0.4	1.71	0.81	0.12	0.16	0.08	5.49	8.31
MILP vs J48	1.13	4.26	6.74	1.63	3.16	9.36	10.32	5.06	6.63
MILP vs Random Forest	1.25	1.52	3.86	0.82	0.72	2.04	2.75	3.81	4.64
MILP vs RBF Network	1.62	7.21	10.87	3.58	8.81	14.83	26.38	1.08	1.38
MILP vs JRip	3.88	6.7	8.7	4.09	3.07	17.11	26.27	4.79	14.95
MILP vs NaiveBayes	2.34	7.29	11.51	4.9	9.43	15.3	31.31	1.42	2.39
MILP vs Logistic	2.46	8.62	12.77	6.25	7.29	14.93	27.49	0.48	0

The accuracy values of MILP approach on each of the data sets is statistically significant than the accuracies of the distance based algorithms HD and ED given in [97] and [100]. Since there are not any existing literature results of these methods for 1189 and 25PDB data sets, *P*-value analysis for these data sets are not available. On the other hand, there is no statistical difference between the CC algorithm and MILP approach for 138, 277 and 498 data sets. However, MILP approach is statistically significant than CC algorithm for the data sets 253, 359, 510 and 2438. There is no statistically significant difference between the accuracy values of SVM given in [109] and [110] and proposed MILP approach. However, the results given in [110] are not consistent with the results achieved by LibSVM and SMO.

Table 4.26 The results of P-test.

Compared Methods	**138**	**253**	**359**	**277**	**498**	**510**	**2438**	**1189**	**25PDB**
	P Test Result	**P Test Result**	**P Test Result**	**P Test Result**	**P Test Result**	**P Test Result**	**P Test Result**	**P Test Result**	**P Test Result**
MILP vs HD	+ +	+ +	+ +	+ +	+ +	+ +	+ +	NA	NA
MILP vs ED	+ +	+ +	+ +	+ +	+ +	+ +	+ +	NA	NA
MILP vs CC	= =	+ +	+ +	= =	= =	+ +	+ +	NA	NA
MILP vs SVM	= =	= =	= =	= =	= =	= =	= =	NA	NA
MILP vs LibSVM	= =	= =	+ +	= =	= =	+ +	+ +	= =	= =
MILP vs SMO	= =	+ +	+ +	+ +	+ +	+ +	+ +	= =	= =
MILP vs NN	NA	NA	NA	= =	= =	NA	NA	NA	NA
MILP vs IB1	= =	= =	= =	= =	= =	= =	= =	+ +	+ +
MILP vs J48	= =	+ +	+ +	= =	+ +	+ +	+ +	+ +	+ +
MILP vs Random Forest	= =	= =	+ +	= =	= =	+ +	+ +	+ +	+ +
MILP vs RBF Network	= =	+ +	+ +	+ +	+ +	+ +	+ +	= =	= =
MILP vs JRip	+ +	+ +	+ +	+ +	+ +	+ +	+ +	+ +	+ +
MILP vs NaiveBayes	+ +	+ +	+ +	+ +	+ +	+ +	+ +	= =	+ +
MILP vs Logistic	+ +	+ +	+ +	+ +	+ +	+ +	+ +	+ +	= =

+ + denotes that the first method is statistically significantly better than the second method. - - represents that the second method is statistically significantly better than the first method. = = indicates that there is no significant difference between the results of the methods. HD: Hamming Distance. ED: Euclidean Distance. CC: Component-coupled. SVM: Support Vector Machines. NN: Neural Networks.

MILP approach is statistically significantly better than the support vector machine algorithms implemented in LibSVM and WEKA for some of the data sets. Similarly, there is no statistically significant difference between the LOO results of Neural Network given in [104] and MILP approach on 277 and 498 data sets. On the other hand, MILP approach is statistically significant than the Neural Network classifier found in WEKA (RBF Network) for most of the data sets. There is no statistically significant difference between the results of IB1 classifier and MILP approach for each of the data sets except 1189 and 25PDB. Surprisingly, IB1 has worse accuracy value with respect to MILP approach for data sets 1189 and 25PDB. Finally, MILP approach has statistically significant accuracy values for the methods J48, Random Forest (RF), JRip, Naïve Bayes (NB) and Logistic for most of the data sets.

In order to compare the existing data classification methods with MILP, some of the ordered *P*-value graphs are shown in Figure 4.1 to Figure 4.6. In Figure 4.1, the ordered P-values of MILP versus LibSVM for each of the nine data sets are shown. For three data sets, the P-values are greater than 2 and very close to 15 which is a considerably high P-value. In general, MILP is preferable since it performs quite well for each of the existing benchmark data sets. However, LibSVM method performs poorly with respect to MILP approach for 3 of the data sets. Hence, we could say that MILP approach is significantly better than LibSVM method in general. We could come up with the same conclusion for IB1 and MILP methods (Figure 4.2). In a similar way, IB1 method performs worse for two of the data sets despite its high efficiency for the rest of the data sets. Thus, MILP approach is statistically better than IB1 method in general. MILP approach is statistically significant than SMO, Logistic and RBF Network algorithms found in WEKA in most of the data sets (Figure 4.3, 4.4 and 4.5). Moreover, the highest P-value for these methods is close to 30 which mean that the difference between the performances of the methods and MILP is highly significant. Finally, proposed MILP approach is statistically significantly better than Random Forest algorithm for half of the data sets (Figure 4.6). For the rest, the difference between the accuracies of two methods is not significant. Moreover, P-values are not very high for Random Forest algorithm compared to the rest of the listed methods in Figure 2. For each of the existing protein folding type benchmark data sets, MILP approach generally achieves high accuracy values and mostly ranks in first three positions with respect to accuracy. Furthermore, MILP is statistically significantly better than the existing data classification methods for protein folding type prediction problems on given nine distinct benchmark data sets.

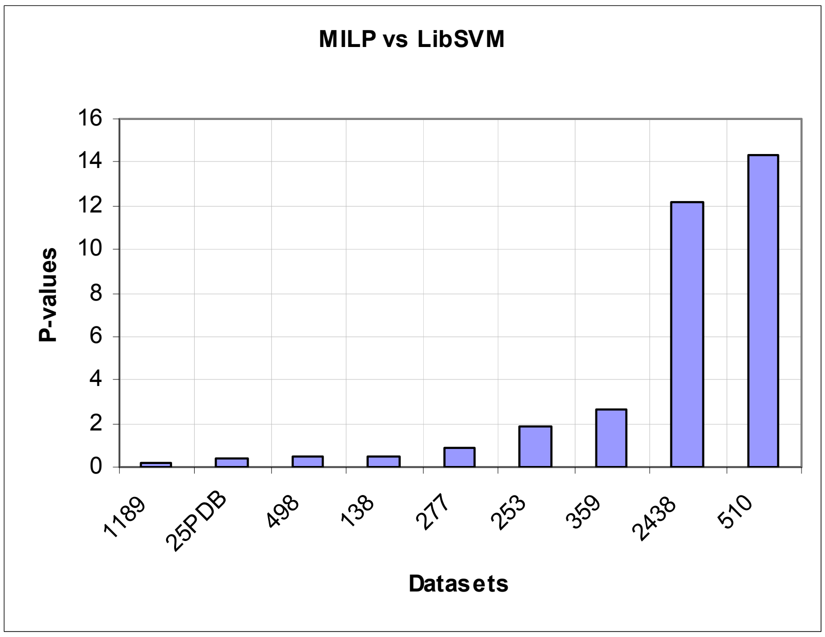

Figure 4.1 P-value graph of MILP versus LibSVM.

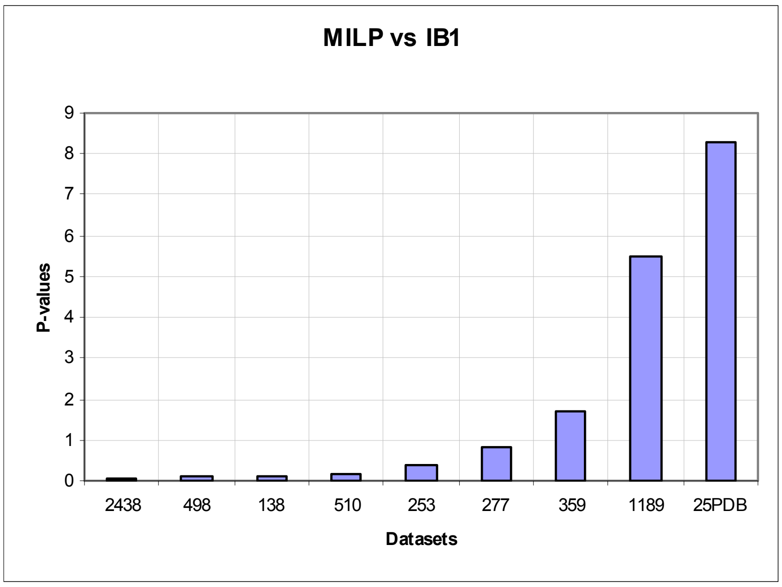

Figure 4.2 P-value graph of MILP versus IB1.

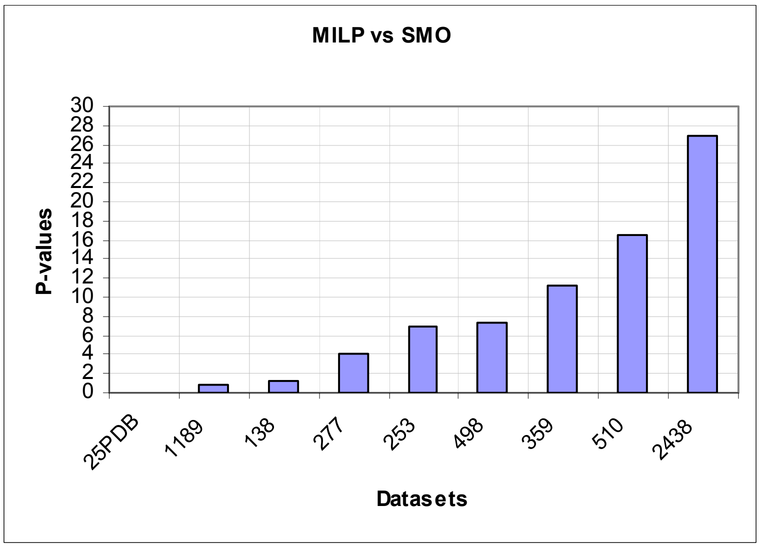

Figure 4.3 P-value graph of MILP versus SMO.

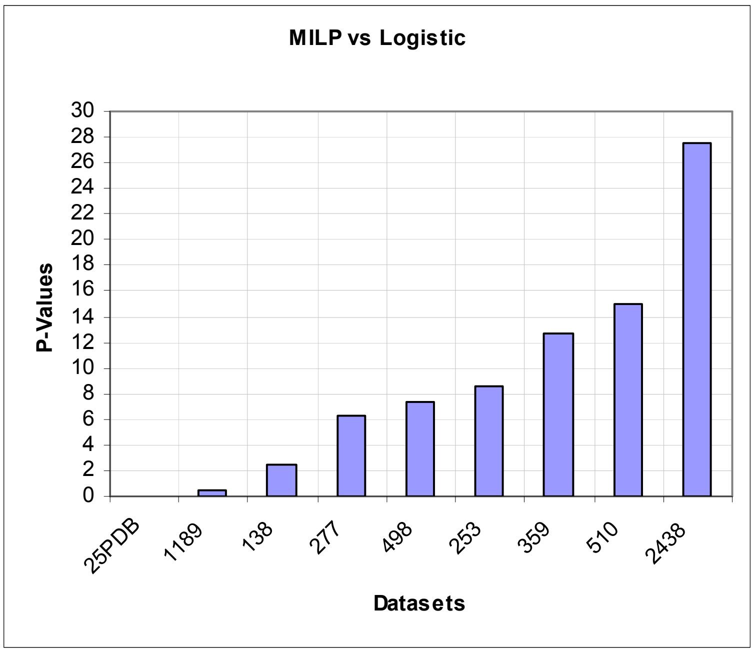

Figure 4.4 P-value graph of MILP versus Logistic.

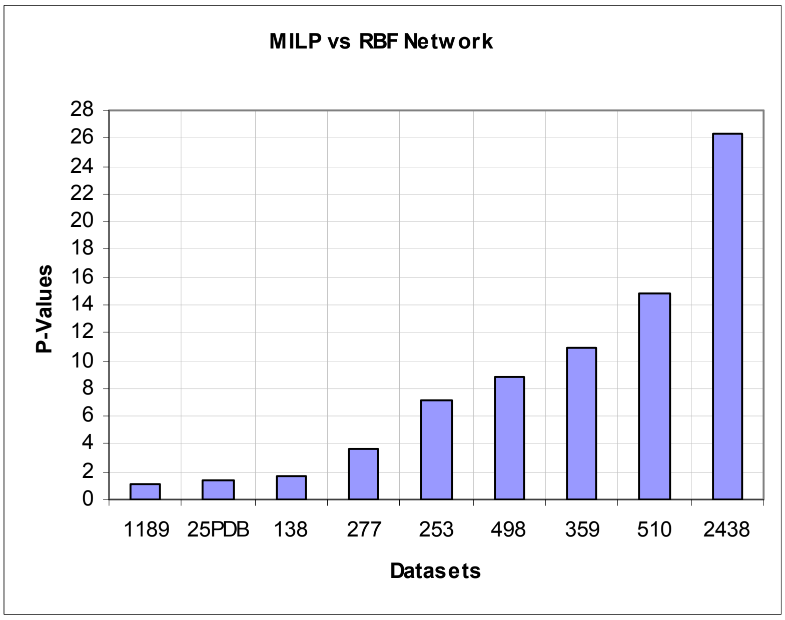

Figure 4.5 P-value graph of MILP versus RBF Network.

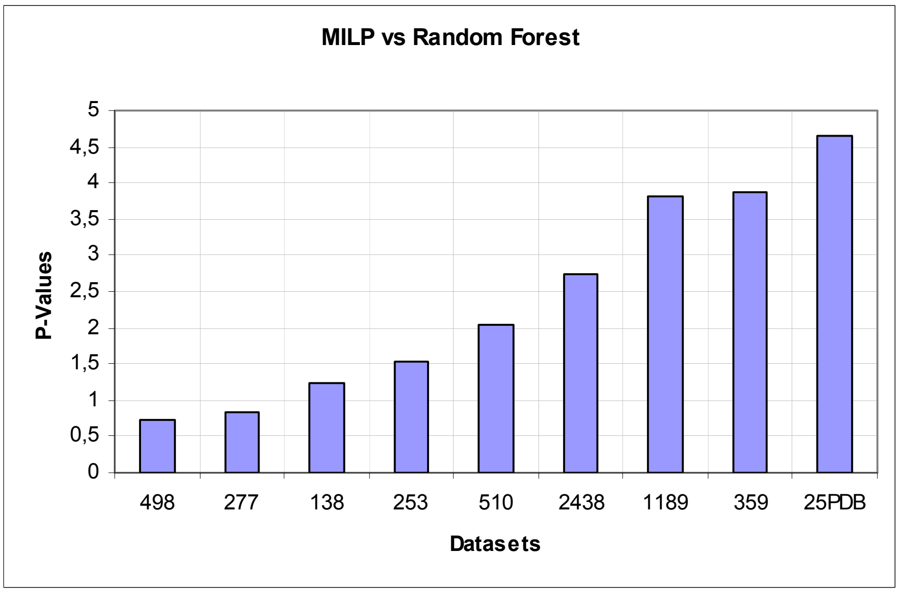

Figure 4.6 P-value graph of MILP versus Random Forest.

4.9 Problematic Instance Analysis

In order to analyze whether there exists any relation between the performance of the proposed approach and the number of problematic instances in the data sets, we investigate the results of these data sets in detail. In Table 4.27, the number of problematic instances is given by average, maximum and minimum values for each of the protein folding type benchmark data sets. As 225 and 1601 data sets are used for training sets for the test sets 510 and 2438 data sets, they do not have any maximum and minimum number of problematic instances. On the other hand, since for the data sets 138, 253, 359, 277 and 498 LOO tests are carried, their problematic instance analyses are comprehensive (Figure 4.7 and 4.11). Furthermore, the number of problematic instances for 1189 and 25PDB data sets change from one run to another as 10FCV results are obtained for them (Figure 4.12 – Figure 4.13).

As it can be observed from the Table 4.27, the number of problematic instances does not affect the performance of the proposed approach. For the same percentage of problematic instances as in 359 and 498 data sets, the proposed approach could achieve the best and second best results for 359 and 498 data sets, respectively. Moreover, for the data sets that have high percentage of problematic instances as 1601 data set, proposed approach could very high accuracy value, 95.88%. Hence, considering only the number of problematic instances could not be sufficient to analyze the difficulty of the data sets.

Table 4.27 Number of problematic instances for each of the protein folding type data sets.

Data Set Name	Accuracy (%)	Accuracy Rank	% of Av. Problematic Instances	Number of Problematic Instances		
				Average	Max.	Min.
138	67.39%	2	53%	73	74	69
253	87.65%	1	65%	164	165	157
359	96.38%	1	65%	233	234	228
277	81.50%	3	60%	167	168	161
498	92.97%	2	65%	322	323	312
225	95.88%	1	78%	179	N/A	N/A
1601	95.08%	1	97%	1554	N/A	N/A
1189	53.30%	1	88%	959	962	956
25PDB	51.82%	3	89%	1486	1489	1482

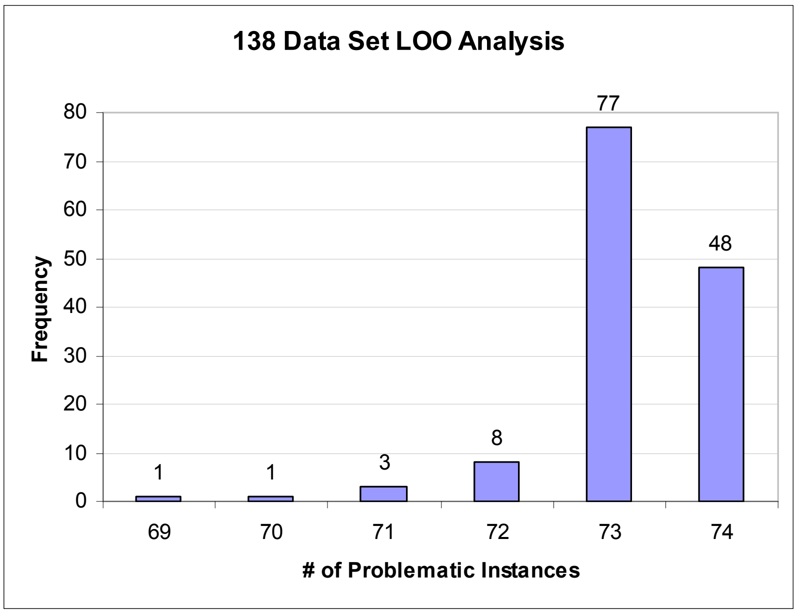

Figure 4.7 The number of problematic instances for 138 data set.

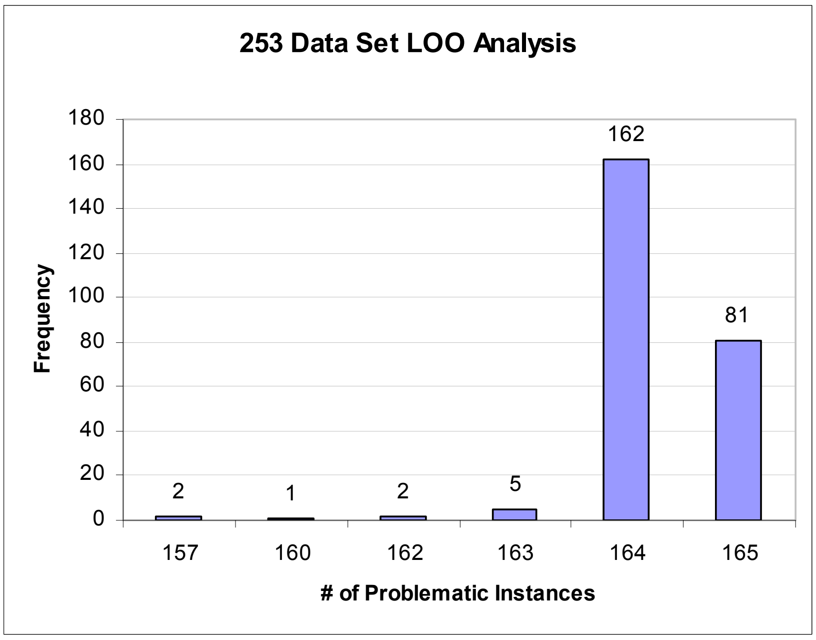

Figure 4.8 The number of problematic instances for 253 data set.

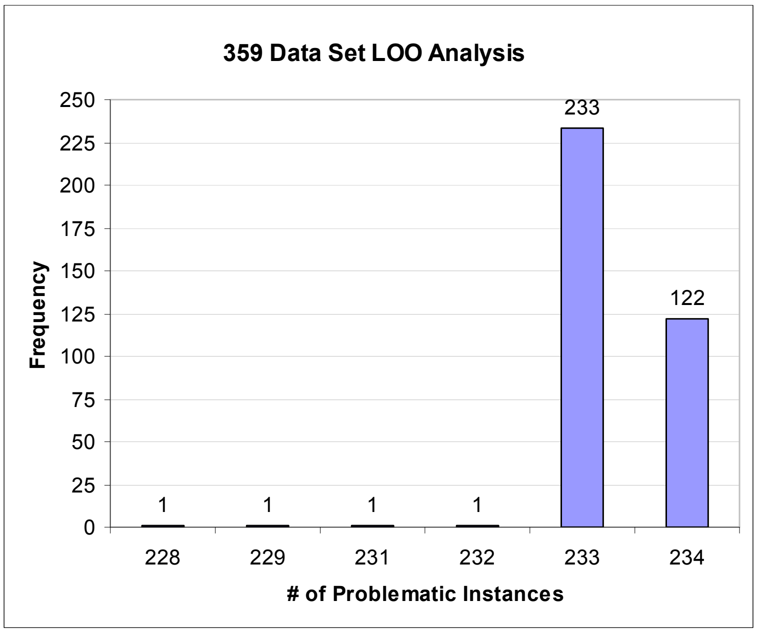

Figure 4.9 The number of problematic instances for 359 data set.

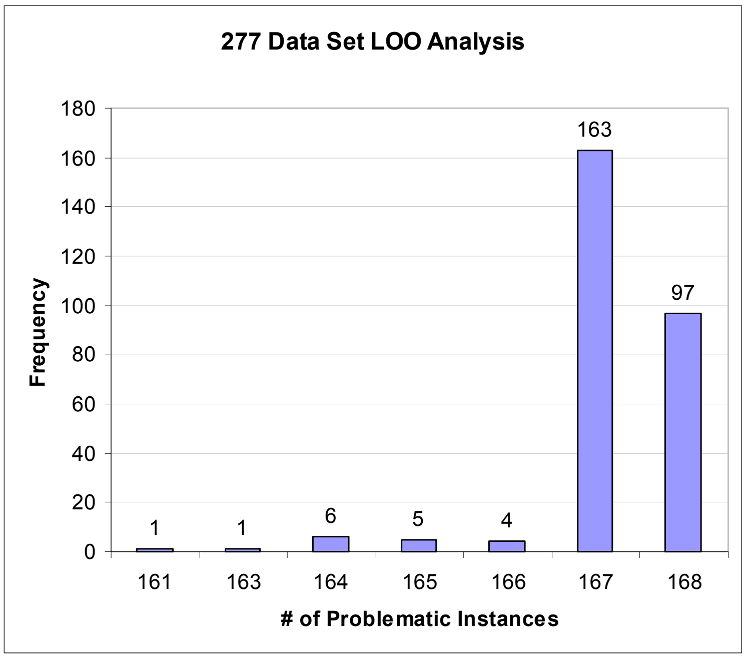

Figure 4.10 The number of problematic instances for 277 data set.

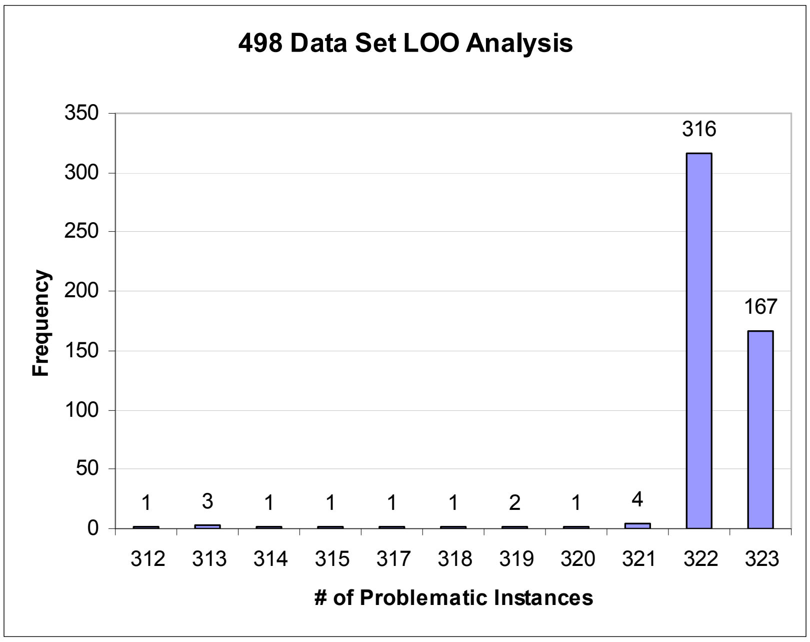

Figure 4.11 The number of problematic instances for 498 data set.

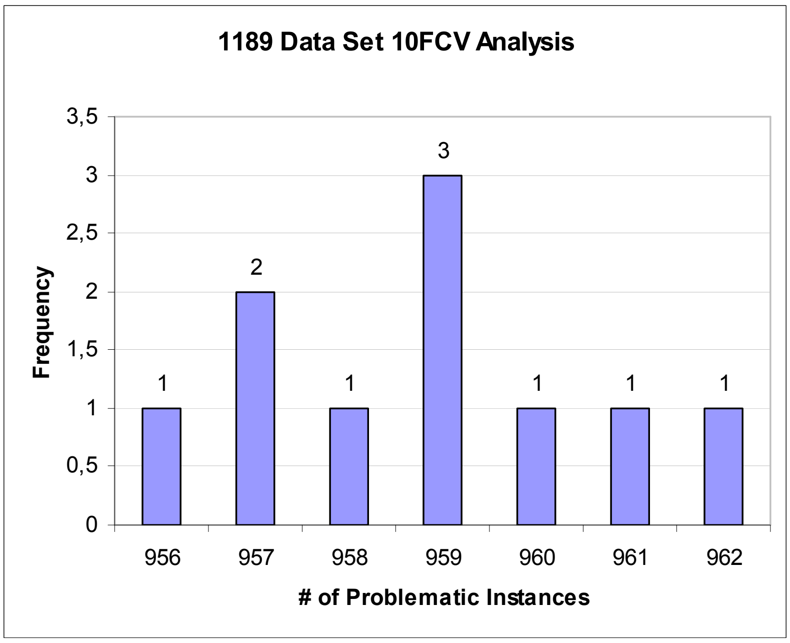

Figure 4.12 The number of problematic instances for 1189 data set.

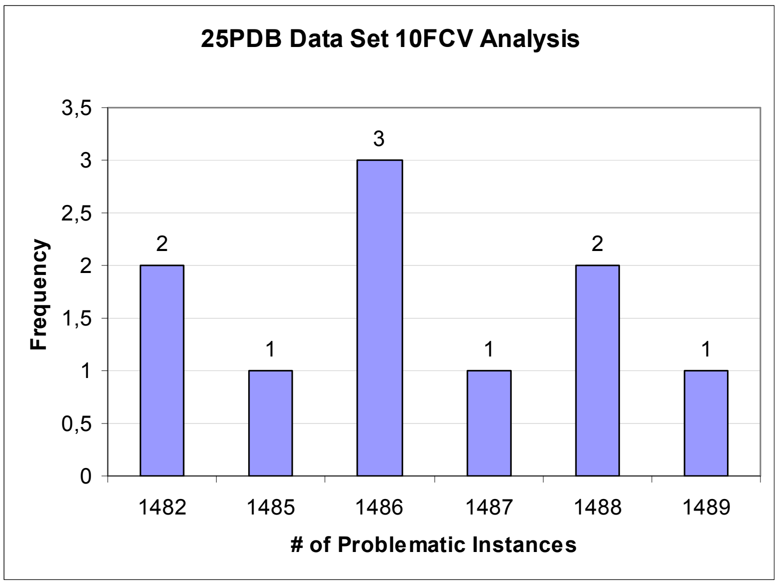

Figure 4.13 The number of problematic instances for 25PDB data set.

Chapter 5

COMPUTATIONAL RESULTS ON UCI REPOSITORY DATA SETS

The performance of proposed three-stage approach is evaluated on eleven UCI repository benchmark data sets [108]. The prediction results and comparisons with other data classification methods are examined in this chapter.

5.1 UCI Repository Data Sets

The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. The archive was created in 1987 by David Aha and fellow graduate students at UC Irvine. Since that time, it has been widely used by students, educators, and researchers all over the world as a primary source of machine learning data sets [108].

In order to observe the performance of the proposed MILP based hyper-box enclosure approach, the following eleven data sets from [108] are tested. First five of them are binary-class data classification data sets and the rest are multi-class data sets.

5.1.1 Johns Hopkins University Ionosphere Database

This database contains the radar data collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. Free electrons in the ionosphere are the targets of this study. “Good” radar returns are those showing evidence of some type of structure in the ionosphere. On the other hand, “Bad” radar returns are those that do not show any evidence and their signals pass through the ionosphere.

Received signals were processed using an autocorrelation function that depends on the time of the pulse and the pulse number. There were 17 pulse numbers for the Goose Bay system. Instances in this database are described by 2 attributes per pulse number, corresponding to the complex values returned by the function resulting from the complex electromagnetic signal. The overall characteristics of the database are given in Table 5.1. This data set is referred as “Ionosphere”.

Table 5.1 Binary-class UCI Repository data sets and their characteristics.

Data Set Name	**# of Attributes**	**# of Classes**	**# of Instances**	**# of Instances in Class 1**	**# of Instances in Class 2**
Ionosphere	34	2	351	225	126
Pima	8	2	768	500	268
Blood	4	2	748	570	178
WDBC	9	2	683	444	239
Liver	6	2	345	200	145

5.1.2 Pima Indians Diabetes Database

This database consists of female patients at least 21 years old who have Pima Indian Heritage. The given 8 properties related to the patients are used to test the diabetes for each one of them. The overall characteristics of the database are given in Table 5.1. This data set is referred as “Pima”.

5.1.3 Blood Transfusion Service Center Data Set

This data set is taken from the donor database of Blood Transfusion Service Center in Hsin-Chu City in Taiwan. 748 donors are randomly selected from the databases with information related to the months since last donation, months since first donation, total blood donated in c.c., total number of donation. The class variable represents whether she/he donated blood in March 2007. The overall characteristics of the data set are given in Table 5.1. This data set is referred as “Blood”.

5.1.4 Wisconsin Diagnostic Breast Cancer (WDBC)

This breast cancer database was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. Using the 9 different information related to the patients, one is trying to find out whether the patient has a breast cancer or not. The overall characteristics of the database are given in Table 5.1. This data set is referred as “WDBC”.

5.1.5 Liver Disorders Data Set

This data set is consists of the records of male individuals with 5 blood test values which are thought to be sensitive to liver disorders that might arise from excessive alcohol consumption. Moreover, each individual have an attribute value related to the number of half-pint equivalents of alcoholic beverages drunk per day. The class variable represents whether he has a liver disorder or not. The overall characteristics of the data set are given in Table 5.1. This data set is referred as “Liver”.

5.1.6 Wine Recognition Data

These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of

13 constituents found in each of the three types of wines. The overall characteristics of the data set are given in Table 5.2. This data set is referred as "Wine".

Table 5.2 Multi-class UCI Repository data sets and their characteristics.

Data Set	#	#	#	# of Instances in each class									
	Att.	Classes	Ins.	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Wine	13	3	178	59	71	48	---	---	---	---	---	---	---
Iris	4	3	150	50	50	50	---	---	---	---	---	---	---
Thyroid	5	3	215	150	35	30	---	---	---	---	---	---	---
Glass	9	6	214	70	76	17	13	9	29	---	---	---	---
Ecoli	7	8	336	143	77	2	2	35	20	5	52	---	---
Yeast	8	10	1484	244	429	463	44	35	51	163	30	20	5

5.1.7 Iris Data Set

Iris data is the best known data set to be found in the pattern recognition literature. The sepal length, sepal width, petal length, and petal width are measured in centimeters on 50 iris specimens from each of three species, *Iris setosa, I. versicolor,* and *I. virginica*. The overall characteristics of the data set are given in Table 5.2. This data set is referred as "Iris".

5.1.8 Thyroid Gland Data

This data set composed of five laboratory tests of patients to predict whether a patient's thyroid to the class euthyroidism, hypothyroidism or hyperthyroidism. The diagnosis (the class label) was based on a complete medical record, including anamnesis, scan, etc. The overall characteristics of the data set are given in Table 5.2. This data set is referred as "Thyroid".

5.1.9 Glass Identification Database

This database is composed of 6 different types of glasses with having some chemical properties to differentiate. The aim is to classify the glasses using the 9 characteristics of instances exist in the data set. The overall characteristics of the data set are given in Table 5.2. This data set is referred as "Glass".

5.1.10 Ecoli Data Set

This data set is composed of proteins with 7 different score values and a localization site. There are 8 different sites that proteins are localized. The overall characteristics of the data set are given in Table 5.2. This data set is referred as "Ecoli".

5.1.11 Yeast Data Set

This data set is also composed of proteins with 8 different score values and a cellular localization site. There are 10 different cellular sites that proteins are localized. The overall characteristics of the data set are given in Table 5.2. This data set is referred as "Yeast".

5.2 Classification Algorithms

In order to compare the results of proposed MILP approach, WEKA classification algorithms J48, RBF Network, Logistic, Naïve Bayes (NB), SMO, Random Forest (RF) and IB1 are studied (Table 5.3). Optimized parameter values of these WEKA classifiers are determined and used to perform the studies on the given data sets. Moreover, well-known support vector machine implementation LibSVM given by [111] is also studied to observe the accuracy values. For each of the data sets, parameters related to SVM algorithm are optimized by performing 10FCV validation with different combinations of cost and gamma values. The optimal values that achieve the highest 10FCV accuracy are used to obtain the results for each data set (Table 5.4).

Table 5.3 Summary of the applied classification algorithms of WEKA.

Classifier	**Reference**	**Short Description**
Naïve Bayes	[112]	• Class for a Naive Bayes classifier using estimator classes. • Numeric estimator precision values are chosen based on analysis of the training data.
RBF Network	[113]	• Class that implements a normalized Gaussian radial basis function network. • It uses the k-means clustering algorithm to provide the basis functions and learns either a logistic regression (discrete class problems) or linear regression (numeric class problems) on top of that. • It standardizes all numeric attributes to zero mean and unit variance.
IB1	[114]	• IB1-type classifier. • Uses a simple distance measure to find the training instance closest to the given test instance, and predict the same class as this training instance. • If multiple instances are the same (smallest) distance to the test instance, the first one found is used.
J48	[115]	• Class for generating an unpruned or a pruned C4.5 decision tree.
Random Forest	[116]	• Decision tree type algorithm • Class for constructing random forests.
JRip	[117]	• This class implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), which is proposed by William W. Cohen as an optimized version of IREP.
SMO	[118]	• Implements John C. Platt's sequential minimal optimization algorithm for training a support vector classifier using polynomial kernels. • Transforms output of SVM into probabilities by applying a standard sigmoid function that is not fitted to the data.
Logistic	[119]	• Class for building a logistic regression model using LogitBoost. • Incorporates attribute selection by fitting simple regression functions in LogitBoost.

Table 5.4 Optimal parameter values of LibSVM for each of the data sets.

Data Sets	**Kernel Type**	**c (Cost)**	**g (Gamma)**
Ionosphere	Radial Basis Function	8192	0.5
Pima	Radial Basis Function	8	0.00003
Blood	Radial Basis Function	2048	0.00012
WDBC	Radial Basis Function	32768	0.00003
Liver	Radial Basis Function	2	0.00012
Wine	Radial Basis Function	8192	0.00003
Iris	Radial Basis Function	2	0.125
Thyroid	Radial Basis Function	512	0.00003
Glass	Radial Basis Function	32768	0.03125
Ecoli	Radial Basis Function	0.5	8
Yeast	Radial Basis Function	0.5	8

5.3 10-Fold Cross-validation Results for Binary-Class Data Sets

For 10-fold cross-validation tests (10FCV), five binary-class data sets including Ionosphere, Pima, Blood, WDBC, and Liver are used. Using these data sets, the proposed three-stage MILP model is solved in GAMS [120] using ILOG CPLEX Solver version 10.0 [121] on a notebook computer with Inter Pentium M 1.73 Ghz Processor and 512 MB of RAM. For each data set, 10 different runs are carried out and average 10FCV results are given.

10FCV test results for Ionosphere data set are given in Table 5.5. The overall accuracy of the proposed model on Ionosphere data set is 94.59%. MILP approach has the second best accuracy value. The highest accuracy value is achieved by LibSVM method with 94.87%. Random Forest classifier has a very close accuracy value to LibSVM and

MILP with 93.45%. On the other hand, the Naïve Bayes classifier has the worst accuracy value for Ionosphere data set. The rest of the classifiers have moderate accuracy values.

Table 5.5 10FCV results for Ionosphere data set.

Methods	**Class-based Accuracy**		**Overall**
	C1	**C2**	**Accuracy**
LibSVM	95.56%	93.65%	**94.87**%
MILP	97.33%	89.68%	94.59%
Random Forest	96.44%	88.10%	93.45%
RBF Network	93.33%	90.48%	92.31%
J48	96.44%	82.54%	91.45%
JRip	91.56%	86.51%	89.74%
Logistic	94.22%	79.37%	88.89%
SMO	96.89%	73.81%	88.60%
IB1	96.89%	67.46%	86.32%
Naïve Bayes	80.44%	86.51%	82.62%

10FCV test results for Pima data set are given in Table 5.6. The highest accuracy value is achieved by proposed MILP approach with 81.25%. SMO and Logistic classifiers are the ones that have the closest accuracy value to the MILP approach's accuracy. As expected, LibSVM has also high accuracy with respect to other classifiers with 76.43%. On the other hand, decision tree based classifiers J48 and Random Forest has low accuracy values compared to the MILP approach. Furthermore, the IB1 classifier has the worst accuracy value for Pima data set.

Table 5.7 gives the 10FCV test results for Blood data set. The highest accuracy value is achieved by proposed MILP approach with 79.95%. The neural network based classifier RBF Network has a very close accuracy value to the MILP. Rule based classifier JRip and decision tree based classifier J48 have also high accuracy value for Blood data set. Support vector machine based classifier LibSVM and SMO have relatively low classification accuracies. The nearest neighborhood based classifier IB1 has the lowest accuracy value, 68.58%, for Blood data set.

Table 5.6 10FCV results for Pima data set.

Methods	**Class-based Accuracy**		**Overall**
	C1	**C2**	**Accuracy**
MILP	62.69%	91.20%	**81.25**%
SMO	54.10%	89.80%	77.34%
Logistic	57.09%	88.00%	77.21%
LibSVM	52.24%	89.40%	76.43%
Naïve Bayes	61.19%	77.80%	76.30%
RBF Network	54.10%	86.80%	75.39%
JRip	57.46%	84.20%	74.87%
J48	59.70%	81.40%	73.83%
Random Forest	61.19%	77.80%	72.01%
IB1	52.99%	79.40%	70.18%

10FCV test results for WDBC data set are given in Table 5.8. The highest accuracy value is achieved by proposed MILP approach with 97.36%. SMO, LibSVM and Logistic classifiers are the ones that have the closest accuracy value to the MILP approach's accuracy. Naïve Bayes, Random Forest, JRip, IB1 and J48 have moderate accuracy values approximately 96%. Furthermore, neural network based classifier RBF Network has the worst accuracy value for WDBC data set, 95.75%.

Table 5.7 10FCV results for Blood data set.

Methods	**Class-based Accuracy**		**Overall**
	C1	**C2**	**Accuracy**
MILP	42.13%	91.75%	**79.95%**
RBF Network	25.84%	96.49%	79.68%
JRip	41.57%	90.53%	78.88%
J48	43.26%	88.60%	77.81%
LibSVM	34.83%	91.23%	77.81%
Logistic	12.36%	97.37%	77.14%
SMO	0.00%	100.00%	76.20%
Naïve Bayes	20.22%	92.63%	75.40%
Random Forest	32.58%	84.74%	72.33%
IB1	37.08%	78.42%	68.58%

Table 5.8 10FCV results for WDBC data set.

Methods	**Class-based Accuracy**		**Overall**
	C1	**C2**	**Accuracy**
MILP	98.87%	94.56%	**97.36%**
SMO	97.30%	96.65%	97.07%
Logistic	97.75%	94.98%	96.78%
LibSVM	97.52%	95.40%	96.78%
Naïve Bayes	95.72%	97.49%	96.34%
Random Forest	97.52%	93.72%	96.19%
JRip	96.40%	95.82%	96.19%
IB1	97.52%	93.31%	96.05%
J48	96.40%	95.40%	96.05%
RBF Network	95.72%	95.82%	95.75%

Table 5.9 10FCV results for Liver data set.

Methods	**Class-based Accuracy**		**Overall Accuracy**
	C1	**C2**	
LibSVM	64.83%	81%	**74.20%**
MILP	65.52%	79%	73.33%
J48	53.10%	80%	68.70%
Logistic	53.10%	79%	68.12%
Random Forest	62.76%	69%	66.38%
JRip	46.90%	77%	64.35%
RBF Network	51.72%	73.50%	64.35%
IB1	56.55%	67.50%	62.90%
SMO	0.69%	100%	58.26%
Naïve Bayes	76.55%	40%	55.36%

10FCV test results for Liver data set are given in Table 5.9. The overall accuracy of the proposed model on Liver data set is 73.33%. MILP approach has the second best accuracy value. The highest accuracy value is achieved by LibSVM method with 74.20%. J48 and Logistic classifiers have a closer accuracy values with 68.70% and 68.12%, respectively. On the other hand, the famous probabilistic classifier Naïve Bayes has the worst accuracy value for Liver data set. The rest of the classifiers have moderate accuracy values ranging from 58% to 66%.

5.4 10-Fold Cross-validation Results for Multi-Class Data Sets

For 10-fold cross-validation tests (10FCV) of multi-class problems, six data sets including Wine, Iris, Thyroid, Glass, Ecoli and Yeast are used. Using these data sets, the proposed three-stage MILP model is solved in GAMS [120] using ILOG CPLEX Solver version 10.0 [121] on a notebook computer with Inter Pentium M 1.73 Ghz Processor and

512 MB of RAM. Similar to two-class case, 10 different runs are carried out and average 10FCV results are given for each of the data sets.

Table 5.10 10FCV results for Wine data set.

Methods	**Class-based Accuracy**			**Overall Accuracy**
	C1	**C2**	**C3**	
SMO	100%	95.77%	100%	**98.31%**
Random Forest	100%	95.77%	100%	**98.31%**
RBF Network	96.61%	100%	97.92%	**98.31%**
Logistic	98.31%	95.77%	97.92%	97.19%
Naïve Bayes	94.92%	95.77%	100%	96.63%
MILP	94.92%	95.77%	93.75%	94.94%
IB1	100%	87.32%	100%	94.94%
JRip	91.53%	94.37%	95.83%	93.82%
J48	98.31%	94.37%	87.50%	93.82%
LibSVM	94.92%	90.14%	91.67%	92.13%

10FCV test results for Wine data set are given in Table 5.10. The overall accuracy of the proposed model on Ionosphere data set is 94.94%. MILP approach has the fourth best accuracy value as distance based classifier IB1. The highest accuracy value is achieved by SMO, Random Forest and RBF Network methods with 98.31%. Logistic and Naïve Bayes classifiers have also higher accuracy values than MILP approach with 97.19% and 96.63%, respectively. On the other hand, the famous decision tree classifier J48 and rule based classifier have the same accuracy value, 93.83%, for Wine data set. Surprisingly, LibSVM has the worst accuracy value, 92.13%, for this data set.

10FCV test results for Iris data set are given in Table 5.11. The overall accuracy of the proposed model on Iris data set is 96%. MILP approach has the second best accuracy value. The highest accuracy value is achieved by LibSVM method with 98%. Logistic, J48, Naïve Bayes and SMO classifiers have the same accuracy value with the MILP approach. On the other hand, Random Forest, IB1 and RBF Network have equal accuracy value, 95.33%, for Iris data set. The lowest accuracy value, 94%, is achieved by the rule based classifier JRip.

Table 5.11 10FCV results for Iris data set.

Methods	**Class-based Accuracy**			**Overall**
	C1	**C2**	**C3**	**Accuracy**
LibSVM	100%	96%	98%	**98%**
MILP	100%	94%	94%	96%
Logistic	100%	92%	96%	96%
J48	98%	94%	96%	96%
Naïve Bayes	100%	96%	92%	96%
SMO	100%	98%	90%	96%
Random Forest	100%	96%	90%	95.33%
IB1	100%	94%	92%	95.33%
RBF Network	100%	92%	94%	95.33%
JRip	100%	90%	92%	94%

Table 5.12 gives the 10FCV test results for Thyroid data set. The highest accuracy value is achieved by proposed MILP approach with 97.21%. The nearest neighborhood based classifier IB1 has the same accuracy value with MILP approach. The famous probabilistic classifier Naïve Bayes and Logistic classifier has the second best results with the accuracy value of 96.74%. RBF Network has 95.35% accuracy and stand at the third order. Random forest and LibSVM has the same accuracy values, 93.95%. The support vector machine based classifier SMO has the lowest accuracy value, 89.77%, for Thyroid data set.

Table 5.12 10FCV results for Thyroid data set.

Methods	**Class-based Accuracy**			**Overall**
	C1	**C2**	**C3**	**Accuracy**
MILP	98.67%	91.43%	93.33%	**97.21%**
IB1	99.33%	94.29%	86.67%	**97.21%**
Naïve Bayes	97.33%	94.29%	86.67%	96.74%
Logistic	100%	57.14%	76.67%	96.74%
RBF Network	98%	97.14%	93.33%	95.35%
Random Forest	94.67%	88.57%	83.33%	93.95%
LibSVM	97.33%	85.71%	86.67%	93.95%
JRip	94.67%	85.71%	93.33%	93.02%
J48	99.33%	82.86%	80.00%	92.09%
SMO	98%	94.29%	96.67%	89.77%

10FCV test results for Glass data set are given in Table 5.13. The overall accuracy of the proposed model on Iris data set is 76.17%. MILP approach has the second best accuracy value. The highest accuracy value is achieved by Random Forest classifier with 77.57%. LibSVM and IB1 have accuracy value greater than 70% and are following the MILP approach. On the other hand, JRip, Logistic, J48 and RBF Network have some how closer accuracies to each other for Glass data set. The lowest accuracy value, 49.53%, is achieved by the probabilistic classifier Naïve Bayes.

Table 5.13 10FCV results for Glass data set.

Methods	**Class-based Accuracy**						**Overall**
	C1	**C2**	**C3**	**C4**	**C5**	**C6**	**Accuracy**
Random Forest	82.86%	78.95%	29.41%	76.92%	88.89%	86.21%	**77.57%**
MILP	75.71%	78.95%	47.06%	76.92%	77.78%	86.21%	76.17%
LibSVM	70%	78.95%	17.65%	76.92%	66.67%	86.21%	71.50%
IB1	77.14%	67.11%	35.29%	76.92%	66.67%	82.76%	70.56%
JRip	61.43%	76.32%	5.88%	69.23%	77.78%	82.76%	66.36%
Logistic	67.14%	67.11%	5.88%	76.92%	88.89%	86.21%	66.36%
J48	71.43%	56.58%	29.41%	84.62%	88.89%	82.76%	65.89%
RBF Network	72.86%	63.16%	11.76%	53.85%	77.78%	89.66%	65.89%
SMO	44.29%	85.53%	0.00%	15.38%	0.00%	86.21%	57.48%
Naïve Bayes	71.43%	19.74%	35.29%	23.08%	88.89%	82.76%	49.53%

10FCV test results for Ecoli data set are given in Table 5.14. The overall accuracy of the proposed model on Iris data set is 86.61%. MILP approach has the second best accuracy value. The highest accuracy value is achieved by LibSVM classifier with 87.50%. Logistic classifier has a very close accuracy value to the MILP approach with 86.31%. As the number of instances in the classes C3 and C4 are very low (Table 5.2), the class-based accuracy values for these classes are 0 for each of the methods. As two instances are not sufficient to capture the class characteristic for this large Ecoli data set, these results are not surprising. On the other hand, Naïve Bayes, RBF Network, J48, SMO, and Random Forest have some how closer and moderate accuracies to each other for Ecoli data set. The lowest accuracy value, 80.36%, is achieved by the instance based classifier IB1 and the rule based classifier JRip.

Table 5.14 10FCV results for Ecoli data set.

Methods	Class-based Accuracy								Overall
	C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**Accuracy**
LibSVM	98.60%	84.42%	0%	0%	62.86%	80%	80%	88.46%	**87.50%**
MILP	97.90%	83.12%	0%	0%	60%	80%	80%	88.46%	86.61%
Logistic	96.50%	84.42%	0%	0%	60%	80%	100%	86.54%	86.31%
Naïve Bayes	95.80%	72.73%	0%	0%	82.86%	90%	60%	84.62%	85.42%
RBF Network	96.50%	80.52%	0%	0%	54.29%	75%	80%	88.46%	84.52%
J48	95.10%	84.42%	0%	0%	60.00%	70%	60%	84.62%	84.23%
SMO	98.60%	83.12%	0%	0%	25.71%	75%	100%	90.38%	83.63%
Random Forest	95.80%	81.82%	0%	0%	45.71%	90%	60%	84.62%	83.63%
IB1	93.01%	72.73%	0%	0%	48.57%	75%	100%	84.62%	80.36%
JRip	95.80%	75.32%	0%	0%	51.43%	75%	20%	78.85%	80.36%

Table 5.15 10FCV results for Yeast data set.

Methods	Class-based Accuracy										Overall
	C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**Accuracy**
MILP	57%	50%	76%	89%	49%	45%	84%	10%	45%	40%	**63%**
LibSVM	55%	50%	74%	82%	49%	37%	78%	3%	35%	20%	60%
RBF Network	56%	53%	64%	80%	63%	29%	80%	0%	45%	100%	59%
Random Forest	61%	57%	60%	75%	49%	35%	80%	0%	25%	20%	59%
Logistic	57%	46%	70%	64%	49%	37%	81%	0%	45%	80%	59%
JRip	50%	53%	64%	73%	49%	35%	80%	0%	55%	80%	58%
Naïve Bayes	61%	40%	70%	61%	69%	39%	80%	0%	45%	40%	58%
SMO	56%	35%	78%	80%	29%	20%	78%	0%	55%	60%	57%
J48	54%	51%	56%	82%	43%	43%	83%	0%	25%	80%	56%
IB1	48%	48%	56%	68%	49%	33%	68%	7%	45%	100%	52%

Table 5.15 gives the 10FCV test results for Yeast data set. The highest accuracy value is achieved by proposed MILP approach with 63%. The support vector machine based classifier LibSVM has the closest accuracy value to MILP approach with 60%. RBF Network, Random forest and Logistic classifiers achieved the third best results, 59%, for Yeast data set. The JRip, Naïve Bayes, SMO and J48 have moderate results with 58%, 58%, 57%, and 56%, respectively. The instance based classifier IB1 has the lowest accuracy value, 52%, for Yeast data set.

5.5 Statistical Analysis of the Results

In order to analyze the results in detail, average sensitivity (SEN), average specificity (SPE), MCC and S values of each of the protein data sets are calculated and examined (Table 5.16 - Table 5.28). The average sensitivity values are same as the overall accuracy values. Hence, each of the tables are arranged so as to show the ordering of the methods based on sensitivity values. The average specificity values are generally significantly lower compared to average sensitivity values for the two-class data sets (Table 5.16 – Table 5.20). On the other hand, this observation is not valid for multi-class problems. For six multi-class benchmark data sets, the average specificity values are significantly higher than average sensitivity values (Table 5.17 – Table 5.28). High average specificity means that the number of under predicted proteins is low. Thus, low accuracy is a result of relatively low sensitivity values. Moreover, as average sensitivity values increases, the difference between average sensitivity and average specificity decreases for multi-class problems. Nevertheless, for two-class benchmark data sets, the difference between average sensitivity and average specificity values decreases, as average sensitivity value decreases.

MCC value gives the strength of relationship between the actual and predicted values. A perfect fit will give a MCC value of 1. For two-class benchmark data sets, MCC and S values are equal to the each other for each one of the classes (Table 5.16 – Table

5.20). Moreover, MCC values are always higher than the S values for both two-class and multi-class data sets. On the other hand, each one of the classes in multi-class benchmark data sets has different MCC and S values (Table 5.17 – Table 5.28). Due to the low accuracy values for Blood and Yeast data sets, MCC and S values are low for each of the classes (Table 5.18, Table 5.27, and Table 5.28). This means that the classifier could not effectively capture the characteristics of that class. As accuracy values of data sets including Ionosphere, WDBC, Wine, Iris, Thyroid and Ecoli are high, the MCC and S values are also high for these data sets (Table 5.16, Table 5.19, Table 5.21, Table 5.22, Table 5.23, Table 5.25, and Table 5.26). Furthermore, the data sets Pima, Liver and Glass have moderate MCC and S values as they have moderate accuracy values (Table 5.17, Table 5.20, and Table 5.24).

Table 5.16 Values of performance measures for the Ionosphere data set.

Classifier	**Average Sensitivity**	**Average Specificity**	**MCC**		**S**	
			C1	**C2**	**C1**	**C2**
LibSVM	94.87%	94.33%	0.89	0.89	0.89	0.89
MILP	94.59%	92.43%	0.88	0.88	0.88	0.88
Random Forest	93.45%	91.09%	0.86	0.86	0.86	0.86
RBF Network	92.31%	91.50%	0.83	0.83	0.83	0.83
J48	91.45%	87.53%	0.81	0.81	0.81	0.81
JRip	89.74%	88.32%	0.78	0.78	0.78	0.78
Logistic	88.89%	84.70%	0.76	0.76	0.75	0.75
SMO	88.60%	82.09%	0.75	0.75	0.74	0.74
IB1	86.32%	78.02%	0.70	0.70	0.68	0.68
Naïve Bayes	82.62%	84.33%	0.65	0.65	0.64	0.64

For a perfect prediction, S value should be equal to 1 and 0 for vice versa. Depending on the characteristics of the data sets such as complexity and dimensionality, the prediction accuracies could be low as in the Blood and Yeast data sets (Table 5.18, Table 5.27, and Table 5.28). Hence, the S values for these data sets are very low. On the other hand, when we observe the results of each data set in overall, each of the classes have higher and lower MCC and S values with respect to the remaining classes. Hence, we could not say that MILP based hyper-box enclosure approach performs rather purely for any of the classes. Depending on the data set characteristics, proposed data classification approach works well for each of the classes.

Table 5.17 Values of performance measures for the Pima data set.

Classifier	Average Sensitivity	Average Specificity	MCC		S	
			C1	C2	C1	C2
MILP	81.25%	72.64%	0.57	0.57	0.56	0.56
SMO	77.34%	66.56%	0.48	0.48	0.47	0.47
Logistic	77.21%	67.88%	0.48	0.48	0.47	0.47
LibSVM	76.43%	65.21%	0.46	0.46	0.45	0.45
Naïve Bayes	76.30%	69.29%	0.47	0.47	0.47	0.47
RBF Network	75.39%	65.51%	0.44	0.44	0.43	0.43
JRip	74.87%	66.79%	0.43	0.43	0.43	0.43
J48	73.83%	67.27%	0.42	0.42	0.42	0.42
Random Forest	72.01%	66.99%	0.39	0.39	0.39	0.39
IB1	70.18%	62.20%	0.33	0.33	0.33	0.33

Table 5.18 Values of performance measures for the Blood data set.

Classifier	**Average Sensitivity**	**Average Specificity**	**MCC**		**S**	
			C1	**C2**	**C1**	**C2**
MILP	79.95%	75.58%	0.39	0.39	0.38	0.38
RBF Network	79.68%	42.65%	0.34	0.34	0.29	0.29
JRip	78.88%	53.22%	0.36	0.36	0.36	0.36
J48	77.81%	54.05%	0.35	0.35	0.34	0.34
LibSVM	77.81%	48.25%	0.31	0.31	0.30	0.30
Logistic	77.14%	32.59%	0.19	0.19	0.13	0.13
SMO	76.20%	23.80%	NA	NA	0.00	0.00
Naive Bayes	75.40%	37.46%	0.18	0.18	0.16	0.16
Random Forest	72.33%	44.99%	0.19	0.19	0.19	0.19
IB1	68.58%	46.92%	0.15	0.15	0.15	0.15

Table 5.19 Values of performance measures for the WDBC data set.

Classifier	**Average Sensitivity**	**Average Specificity**	**MCC**		**S**	
			C1	**C2**	**C1**	**C2**
MILP	97.36%	96.07%	0.94	0.94	0.94	0.94
SMO	97.07%	96.88%	0.94	0.94	0.94	0.94
Logistic	96.78%	95.95%	0.93	0.93	0.93	0.93
LibSVM	96.78%	96.14%	0.93	0.93	0.93	0.93
Naïve Bayes	96.34%	96.87%	0.92	0.92	0.92	0.92
Random Forest	96.19%	95.05%	0.92	0.92	0.92	0.92
JRip	96.19%	96.02%	0.92	0.92	0.92	0.92
IB1	96.05%	94.78%	0.91	0.91	0.91	0.91
J48	96.05%	95.75%	0.91	0.91	0.91	0.91
RBF Network	95.75%	95.78%	0.91	0.91	0.91	0.91

Table 5.20 Values of performance measures for the Liver data set.

Classifier	**Average Sensitivity**	**Average Specificity**	**MCC**		**S**	
			C1	**C2**	**C1**	**C2**
LibSVM	74.20%	71.62%	0.47	0.47	0.46	0.46
MILP	73.33%	71.18%	0.45	0.45	0.45	0.45
J48	68.70%	64.41%	0.35	0.35	0.34	0.34
Logistic	68.12%	63.99%	0.33	0.33	0.33	0.33
Random Forest	66.38%	65.38%	0.32	0.32	0.32	0.32
JRip	64.35%	59.55%	0.25	0.25	0.25	0.25
RBF Network	64.35%	60.88%	0.26	0.26	0.26	0.26
IB1	62.90%	61.15%	0.24	0.24	0.24	0.24
SMO	58.26%	42.43%	0.06	0.06	0.01	0.01
Naïve Bayes	55.36%	61.19%	0.17	0.17	0.15	0.15

Table 5.21 Values of performance measures for the Wine data set.

Classifier	**Average Sensitivity**	**Average Specificity**	**MCC**			**S**		
			C1	**C2**	**C3**	**C1**	**C2**	**C3**
RBF Network	98.31%	98.88%	0.97	0.97	0.99	0.96	0.97	0.96
SMO	98.31%	99.31%	0.99	0.97	0.97	0.96	0.96	0.96
Random Forest	98.31%	99.31%	0.99	0.97	0.97	0.96	0.96	0.96
Logistic	97.19%	98.56%	0.97	0.94	0.96	0.94	0.94	0.93
Naïve Bayes	96.63%	98.26%	0.96	0.93	0.96	0.93	0.93	0.92
IB1	94.94%	97.78%	0.94	0.90	0.95	0.90	0.89	0.89
MILP	94.94%	97.16%	0.94	0.91	0.93	0.89	0.90	0.88
J48	93.82%	96.44%	0.94	0.88	0.90	0.87	0.87	0.85
JRip	93.82%	96.44%	0.88	0.88	0.96	0.86	0.87	0.86
LibSVM	92.13%	95.82%	0.91	0.85	0.88	0.83	0.84	0.81

Table 5.22 Values of performance measures for the Iris data set.

Classifier	**Average Sensitivity**	**Average Specificity**	**MCC**			**S**		
			C1	**C2**	**C3**	**C1**	**C2**	**C3**
LibSVM	98%	99%	1.00	0.95	0.96	0.96	0.95	0.96
Naïve Bayes	96%	98%	1.00	0.91	0.91	0.92	0.91	0.91
Logistic	96%	98%	1.00	0.91	0.91	0.92	0.91	0.91
SMO	96%	98%	1.00	0.91	0.91	0.92	0.91	0.91
J48	96%	98%	0.98	0.91	0.93	0.92	0.91	0.91
MILP	96%	98%	1.00	0.91	0.91	0.92	0.91	0.91
Random Forest	95.33	97.67%	1.00	0.90	0.89	0.91	0.90	0.89
IB1	95.33	97.67%	1.00	0.90	0.89	0.91	0.90	0.89
RBF Network	95.33	97.67%	1.00	0.89	0.90	0.91	0.89	0.90
JRip	94%	97%	0.98	0.86	0.88	0.88	0.86	0.87

Table 5.23 Values of performance measures for the Thyroid data set.

Classifier	**Average Sensitivity**	**Average Specificity**	**MCC**			**S**		
			C1	**C2**	**C3**	**C1**	**C2**	**C3**
MILP	97.21%	96.55%	0.93	0.97	0.93	0.93	0.91	0.89
IB1	97.21%	96.52%	0.93	0.95	0.94	0.93	0.91	0.89
Naïve Bayes	96.74%	93.48%	0.92	0.97	0.90	0.92	0.89	0.87
Logistic	96.74%	94.47%	0.92	0.93	0.94	0.92	0.89	0.88
RBF Network	95.35%	93.23%	0.89	0.93	0.88	0.89	0.85	0.82
Random Forest	93.95%	90.02%	0.85	0.89	0.86	0.85	0.81	0.79
LibSVM	93.95%	87.04%	0.86	0.89	0.86	0.85	0.80	0.78
JRip	93.02%	91.81%	0.84	0.83	0.90	0.84	0.78	0.78
J48	92.09%	89.66%	0.81	0.85	0.84	0.81	0.76	0.74
SMO	89.77%	76.39%	0.76	0.73	0.86	0.73	0.64	0.69

Table 5.24 Values of performance measures for the Glass data set.

Classifier	SEN	SPE	MCC						S					
			C1	C2	C3	C4	C5	C6	C1	C2	C3	C4	C5	C6
Random Forest	78%	91%	0.67	0.63	0.32	0.75	0.94	0.87	0.59	0.58	0.39	0.50	0.50	0.58
MILP	76%	91%	0.63	0.63	0.42	0.72	0.82	0.87	0.56	0.57	0.41	0.48	0.48	0.57
LibSVM	71%	89%	0.54	0.59	0.13	0.72	0.74	0.83	0.48	0.52	0.32	0.46	0.45	0.53
IB1	71%	89%	0.59	0.51	0.27	0.66	0.69	0.83	0.51	0.47	0.36	0.45	0.44	0.52
JRip	66%	85%	0.45	0.45	0.08	0.63	0.68	0.85	0.41	0.41	0.33	0.42	0.42	0.49
Logistic	66%	86%	0.43	0.43	0.03	0.69	0.88	0.85	0.40	0.41	0.32	0.43	0.44	0.49
J48	66%	87%	0.48	0.39	0.21	0.74	0.79	0.81	0.43	0.37	0.34	0.44	0.43	0.48
RBF Network	66%	86%	0.47	0.44	0.08	0.50	0.88	0.80	0.43	0.41	0.32	0.40	0.43	0.49
SMO	57%	78%	0.30	0.33	NA	0.24	NA	0.84	0.30	0.28	0.32	0.34	0.34	0.44
Naïve Bayes	50%	82%	0.21	0.11	0.14	0.16	0.75	0.79	0.22	0.18	0.27	0.30	0.35	0.38

Table 5.25 Values of performance measures I for the Ecoli data set.

Classifier	SEN	SPE	MCC							
			C1	C2	C3	C4	C5	C6	C7	C8
SMO	87.50%	96.14%	0.94	0.75	NA	0.00	0.63	0.89	0.89	0.86
Naïve Bayes	86.61%	95.81%	0.93	0.73	NA	0.00	0.61	0.86	0.89	0.86
MILP	86.31%	96.37%	0.92	0.78	NA	NA	0.60	0.81	0.91	0.83
Random Forest	85.42%	96.35%	0.91	0.74	0.00	NA	0.67	0.87	0.77	0.80
LibSVM	84.52%	96.01%	0.93	0.71	0.00	NA	0.54	0.83	0.80	0.83
Logistic	84.23%	96.04%	0.91	0.76	NA	NA	0.59	0.68	0.59	0.81
IB1	83.63%	94.81%	0.92	0.69	NA	NA	0.36	0.86	0.70	0.84
J48	83.63%	95.39%	0.91	0.72	NA	NA	0.47	0.83	0.77	0.78
JRip	80.36%	94.59%	0.86	0.64	NA	NA	0.43	0.80	0.91	0.76
RBF Network	80.36%	93.87%	0.85	0.71	NA	NA	0.44	0.78	0.17	0.77

Table 5.26 Values of performance measures II for the Ecoli data set.

Classifier	**SEN**	**SPE**	**S**							
			C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**
SMO	87.50%	96.14%	0.79	0.70	0.45	0.45	0.56	0.59	0.50	0.69
Naïve Bayes	86.61%	95.81%	0.78	0.68	0.45	0.45	0.54	0.57	0.50	0.68
MILP	86.31%	96.37%	0.77	0.69	0.45	0.43	0.53	0.56	0.51	0.67
Random Forest	85.42%	96.35%	0.76	0.66	0.44	0.44	0.57	0.58	0.48	0.65
LibSVM	84.52%	96.01%	0.75	0.65	0.44	0.43	0.49	0.55	0.48	0.65
Logistic	84.23%	96.04%	0.74	0.66	0.45	0.43	0.52	0.52	0.47	0.64
IB1	83.63%	94.81%	0.74	0.63	0.45	0.45	0.38	0.54	0.48	0.64
J48	83.63%	95.39%	0.73	0.64	0.45	0.45	0.45	0.55	0.47	0.62
JRip	80.36%	94.59%	0.69	0.57	0.44	0.43	0.42	0.52	0.48	0.59
RBF Network	80.36%	93.87%	0.68	0.60	0.44	0.43	0.43	0.51	0.42	0.59

Table 5.27 Values of performance measures I for the Yeast data set.

Classifier	**SEN**	**SPE**	**MCC**									
			C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**
LibSVM	63.14%	86.56%	0.53	0.40	0.40	0.78	0.59	0.49	0.81	0.31	0.55	0.63
MILP	60.38%	85.51%	0.49	0.36	0.35	0.73	0.59	0.42	0.76	0.18	0.46	0.45
Logistic	59.10%	86.04%	0.51	0.33	0.32	0.69	0.57	0.34	0.71	-0.01	0.55	0.91
Naïve Bayes	58.89%	86.48%	0.48	0.34	0.33	0.70	0.51	0.37	0.73	-0.02	0.42	0.25
RBF Network	58.63%	85.68%	0.48	0.33	0.32	0.61	0.51	0.37	0.75	-0.01	0.57	0.73
J48	58.09%	85.08%	0.48	0.34	0.28	0.68	0.54	0.39	0.71	-0.01	0.63	0.89
SMO	57.61%	86.29%	0.50	0.31	0.32	0.59	0.48	0.34	0.75	-0.01	0.55	0.63
Random Forest	57.01%	84.14%	0.46	0.29	0.32	0.69	0.44	0.28	0.75	NA	0.63	0.77
IB1	55.86%	85.53%	0.39	0.27	0.27	0.78	0.45	0.40	0.74	-0.02	0.39	0.80
JRip	52.29%	85.07%	0.33	0.23	0.24	0.66	0.43	0.31	0.64	0.04	0.39	1.00

Table 5.28 Values of performance measures II for the Yeast data set.

Classifier	**SEN**	**SPE**	**S**									
			C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**
LibSVM	63.14%	86.56%	0.41	0.37	0.37	0.41	0.39	0.39	0.46	0.38	0.39	0.39
MILP	60.38%	85.51%	0.39	0.34	0.33	0.39	0.38	0.37	0.43	0.36	0.38	0.38
Logistic	59.10%	86.04%	0.39	0.32	0.32	0.39	0.38	0.36	0.42	0.36	0.37	0.37
Naïve Bayes	58.89%	86.48%	0.38	0.33	0.33	0.38	0.37	0.36	0.42	0.36	0.37	0.37
RBF Network	58.63%	85.68%	0.38	0.32	0.31	0.38	0.37	0.36	0.42	0.36	0.37	0.37
J48	58.09%	85.08%	0.37	0.33	0.29	0.38	0.37	0.36	0.41	0.35	0.37	0.37
SMO	57.61%	86.29%	0.38	0.31	0.31	0.37	0.36	0.35	0.42	0.35	0.37	0.37
Random Forest	57.01%	84.14%	0.37	0.29	0.29	0.38	0.36	0.35	0.41	0.35	0.37	0.36
IB1	55.86%	85.53%	0.34	0.29	0.29	0.38	0.36	0.35	0.41	0.34	0.36	0.36
JRip	52.29%	85.07%	0.31	0.26	0.26	0.35	0.34	0.33	0.37	0.33	0.34	0.35

In order to evaluate if there is any statistical significant difference between the existing and proposed data classification approaches tested on the same data sets, *P*-value (paired test) analysis are carried out. The results of *P*-value test for two-class benchmark data sets are given in Table 5.29 and Table 5.30. There is no statistical significant difference between the results LibSVM and MILP approach on two-class problems except Pima data set. MILP approach is statistically significant than the LibSVM method for Pima data set. On the other hand, there is no significant difference between the results of SMO, support vector implementation of WEKA, and MILP approach for the data sets Pima, Blood and WDBC. However, MILP approach is statistically significant than the accuracies of the SMO classifier on Ionosphere and Liver data sets. The accuracy values of MILP approach on each of the data sets is statistically significant than the accuracies of the distance based algorithm IB1 except the WDBC data set. There is no statistically significant difference between the accuracies of the decision tree based classifier J48 and

MILP approach except the Pima data set. MILP approach is statistically significant than the J48 classifier for Pima data set. There is no statistical significant difference between the results of Random Forest and RBF Network classifier with the MILP approach except the two data sets. MILP approach is statistically significant than the results of Random Forest classifier for Pima and Blood data sets. Furthermore, MILP approach is statistically significant than the results of RBF Network classifier for Pima and Liver data sets. Fro most of the data sets; MILP approach has statistically significant accuracy values from the classifiers JRip and Naïve Bayes. Finally, there is no significant difference between the results of MILP approach and Logistic classifier for each of the data sets except Ionosphere. When we observe the results from the data sets one by one, each of the methods applied to WDBC data set have statistically equivalent results. For the rest of the data sets, MILP approach statistically dominates the results of some of the classifiers.

Table 5.29 The results of P-value analyses for two-class data sets.

Compared Methods	**Ionosphere**	**Pima**	**Blood**	**WDBC**	**Liver**
	P Value	**P Value**	**P Value**	**P Value**	**P Value**
MILP vs LibSVM	0.17	2.31	1.02	0.64	0.26
MILP vs SMO	2.86	1.89	1.75	0.33	4.17
MILP vs IB1	3.73	5.06	5.03	1.36	2.94
MILP vs J48	1.63	3.49	1.02	1.36	1.34
MILP vs Random Forest	0.64	4.28	3.46	1.22	1.99
MILP vs RBF Network	1.22	2.79	0.13	1.63	2.55
MILP vs JRip	2.39	3.02	0.51	1.22	2.55
MILP vs Naïve Bayes	4.99	2.37	2.11	1.08	4.93
MILP vs Logistic	2.74	1.95	1.32	0.64	1.50

Table 5.30 The results of P-test for two-class data sets.

Compared Methods	Ionosphere	Pima	Blood	WDBC	Liver
	P Test Result	P Test Result	P Test Result	P Test Result	P Test Result
MILP vs LibSVM	= =	+ +	= =	= =	= =
MILP vs SMO	+ +	= =	= =	= =	+ +
MILP vs IB1	+ +	+ +	+ +	= =	+ +
MILP vs J48	= =	+ +	= =	= =	= =
MILP vs Random Forest	= =	+ +	+ +	= =	= =
MILP vs RBF Network	= =	+ +	= =	= =	+ +
MILP vs JRip	+ +	+ +	= =	= =	+ +
MILP vs Naïve Bayes	+ +	+ +	+ +	= =	+ +
MILP vs Logistic	+ +	= =	= =	= =	= =

+ + denotes that the first method is statistically significantly better than the second method. - - represents that the second method is statistically significantly better than the first method. = = indicates that there is no significant difference between the results of the methods.

The results of *P*-value test for multi-class benchmark data sets are given in Table 5.31 and Table 5.32. There is no statistical significant difference between the results LibSVM and MILP approach on each one of the multi-class benchmark problems. On the other hand, MILP approach is statistically significant than the accuracies of the SMO classifier on half of the data sets (Thyroid, Glass and Yeast) and there is no significant difference between the results of SMO and MILP approach on half of the data sets (Wine, Iris and Ecoli). There is no statistically significant difference between the performances of the methods IB1 and MILP on each of the multi-class data sets except Ecoli and Yeast. The result of MILP approach is significantly better than the IB1 classifier fro the data sets

Ecoli and Yeast. Similar to SMO classifier, MILP approach is statistically significant than the results of the J48 classifier on half of the data sets (Thyroid, Glass and Yeast) and there is no significant difference between the results of J48 and MILP approach on half of the data sets (Wine, Iris and Ecoli).

Table 5.31 The results of P-value analyses for multi-class data sets.

Compared Methods	**Wine**	**Iris**	**Thyroid**	**Glass**	**Ecoli**	**Yeast**
	P Value	**P Value**	**P Value**	**P Value**	**P Value**	**P Value**
MILP vs LibSVM	1.08	1.02	1.64	1.10	0.34	1.55
MILP vs SMO	1.76	0	3.13	4.11	1.08	3.41
MILP vs IB1	0	0.28	0	1.31	2.18	5.98
MILP vs J48	0.46	0	2.36	2.34	0.88	4.04
MILP vs Random Forest	1.76	0.28	1.64	0.34	1.08	2.37
MILP vs RBF Network	1.76	0.28	1.02	2.34	0.77	2.26
MILP vs JRip	0.46	0.79	2.01	2.24	2.18	2.82
MILP vs Naïve Bayes	0.79	0	0.28	5.70	0.45	3.08
MILP vs Logistic	1.09	0	0.28	2.24	0.11	2.52

The performances of the methods Random Forest and MILP are not statistically significant for each one of the data sets except Yeast. For the Yeast data set, MILP approach is significantly better than the classifier Random Forest. The neural network based classifier RBF Network and MILP approach do not have statistically significant results for each one of the data sets except Glass and Yeast. However, the performance of MILP approach is statistically significant than the RBF Network classifier for the data sets Glass and Yeast. The accuracy values of MILP approach on each of the data sets is statistically significant than the accuracies of the rule based classifier JRip except the data sets Wine and Iris. Similar to RBF Network classifier, Naïve Bayes and Logistic classifiers have no statistically significant results than the MILP approach for the data sets Glass and Yeast. On the other hand, the performances of the methods MILP and Naïve Bayes and Logistic classifiers are not significant in statistical manner for the rest of the multi-class benchmark data sets.

Table 5.32 The results of P-test for multi-class data sets.

Compared Methods	**Wine**	**Iris**	**Thyroid**	**Glass**	**Ecoli**	**Yeast**
	P Test Result	**P Test Result**	**P Test Result**	**P Test Result**	**P Test Result**	**P Test Result**
MILP vs LibSVM	= =	= =	= =	= =	= =	= =
MILP vs SMO	= =	= =	+ +	+ +	= =	+ +
MILP vs IB1	= =	= =	= =	= =	+ +	+ +
MILP vs J48	= =	= =	+ +	+ +	= =	+ +
MILP vs Random Forest	= =	= =	= =	= =	= =	+ +
MILP vs RBF Network	= =	= =	= =	+ +	= =	+ +
MILP vs JRip	= =	= =	+ +	+ +	+ +	+ +
MILP vs Naïve Bayes	= =	= =	= =	+ +	= =	+ +
MILP vs Logistic	= =	= =	= =	+ +	= =	+ +

+ + denotes that the first method is statistically significantly better than the second method. - - represents that the second method is statistically significantly better than the first method. = = indicates that there is no significant difference between the results of the methods.

In order to compare the existing data classification methods with MILP, some of the ordered *P*-value graphs are shown in Figure 5.1 to Figure 5.9. In Figure 5.1, the ordered P-values of MILP versus LibSVM for each of the eleven data sets are shown. For only one data set, the P-value is greater than 2. In general, MILP is preferable since it performs quite well for each of the existing benchmark data sets. However, LibSVM method performs poorly with respect to MILP approach for one of the data sets (Pima). Hence, we could say that MILP approach is significantly better than LibSVM method in general. For more than half of the data sets, MILP approach is statistically significant than SMO, IB1, JRip, and Naïve Bayes classifiers (Figure 5.2, Figure 5.3, Figure 5.7, Figure 5.8). Thus, MILP approach is statistically better than these 4 methods in general. For three data sets, the P-values are greater than 2 for the classifiers Random Forest and Logistic (Figure 5.5 and Figure 5.9). For the rest of the data sets, the difference between the accuracies of two methods is not significant. Finally, proposed MILP approach is statistically significantly better than J48 and RBF Network classifiers for four of the data sets (Figure 5.4 and Figure 5.6).

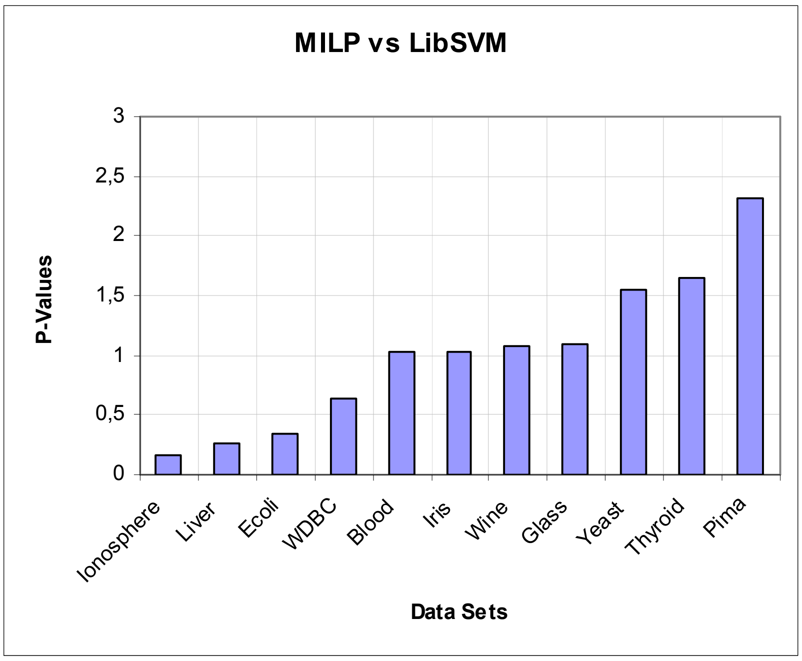

Figure 5.1 P-value graph of MILP versus LibSVM for UCI Benchmark data sets.

For each of the eleven UCI Repository data sets, MILP approach generally achieves high accuracy values and mostly ranks in first three positions with respect to accuracy. Furthermore, MILP is statistically significantly better than the existing data classification methods for these benchmark problems on given eleven distinct benchmark data sets.

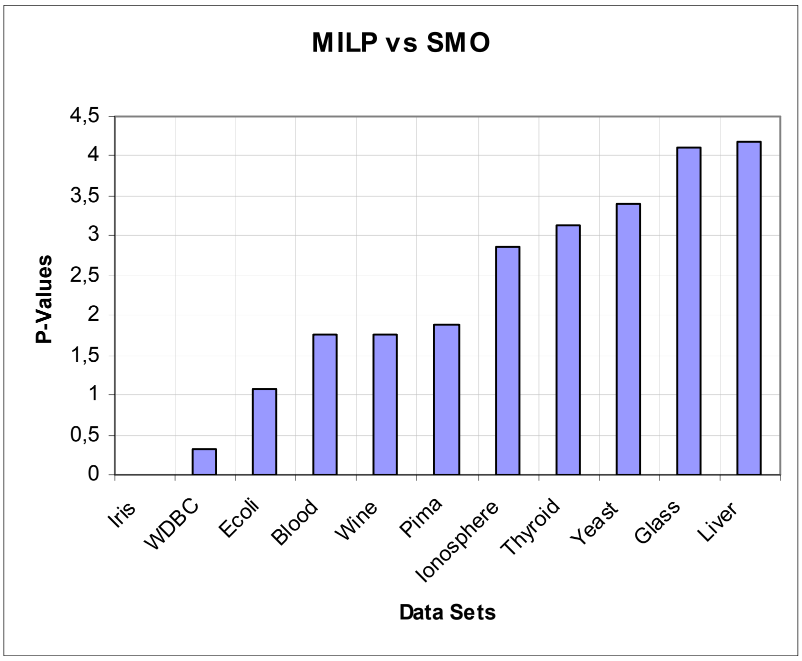

Figure 5.2 P-value graph of MILP versus SMO for UCI Benchmark data sets.

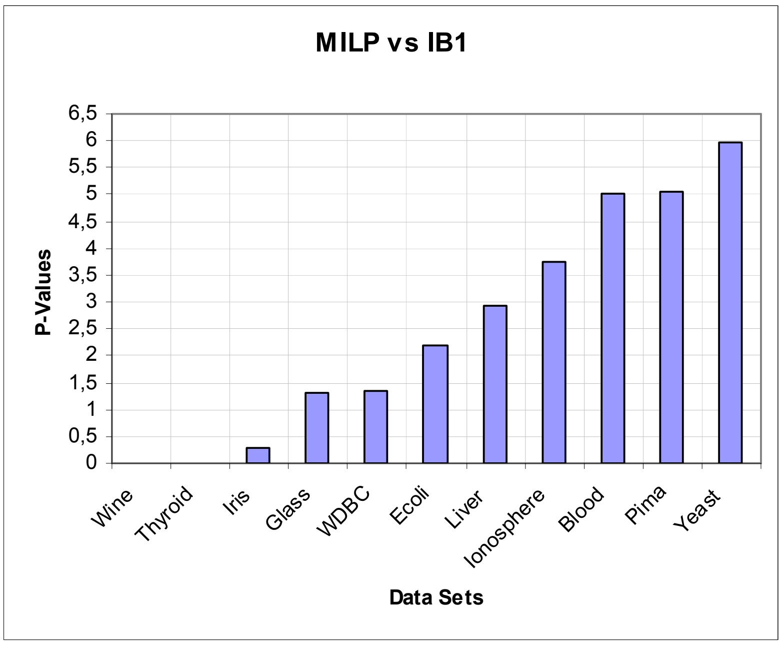

Figure 5.3 P-value graph of MILP versus IB1 for UCI Benchmark data sets.

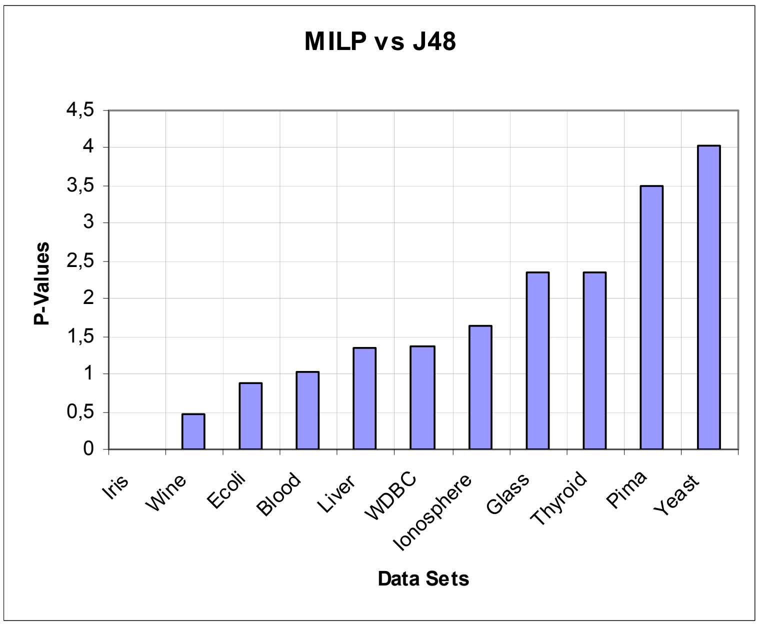

Figure 5.4 P-value graph of MILP versus J48 for UCI Benchmark data sets.

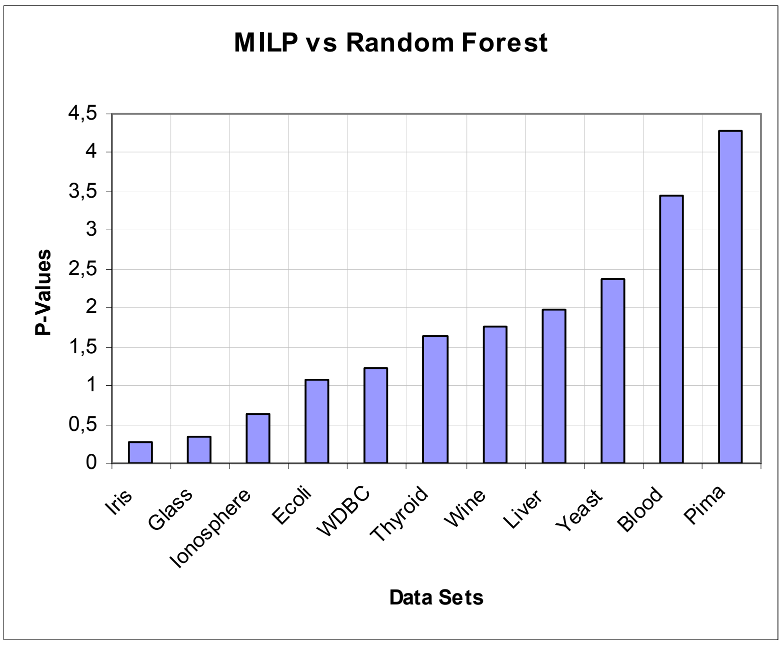

Figure 5.5 P-value graph of MILP versus Random Forest for UCI Benchmark data sets.

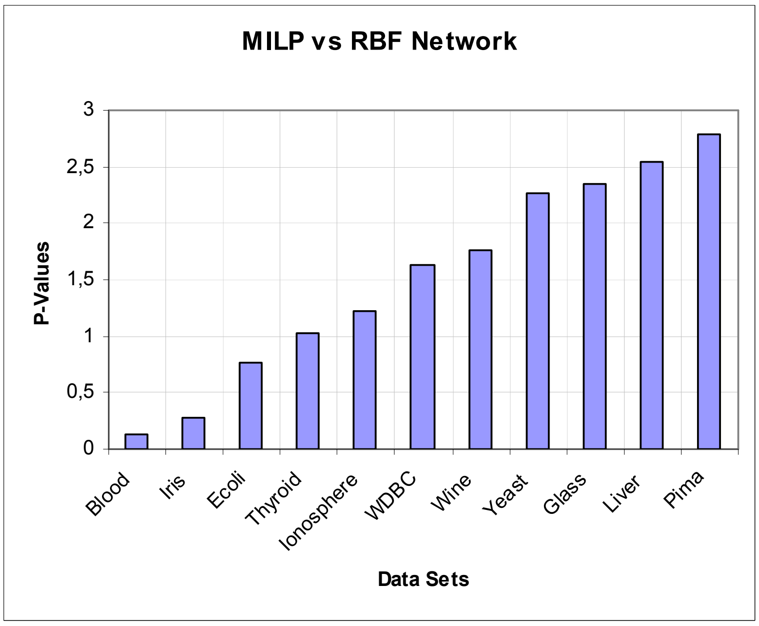

Figure 5.6 P-value graph of MILP versus RBF Network for UCI Benchmark data sets.

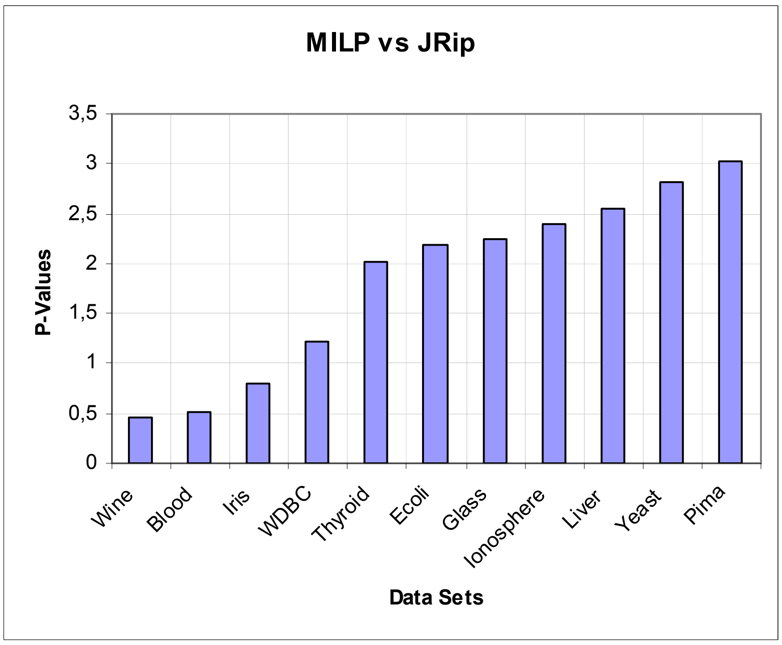

Figure 5.7 P-value graph of MILP versus JRip for UCI Benchmark data sets.

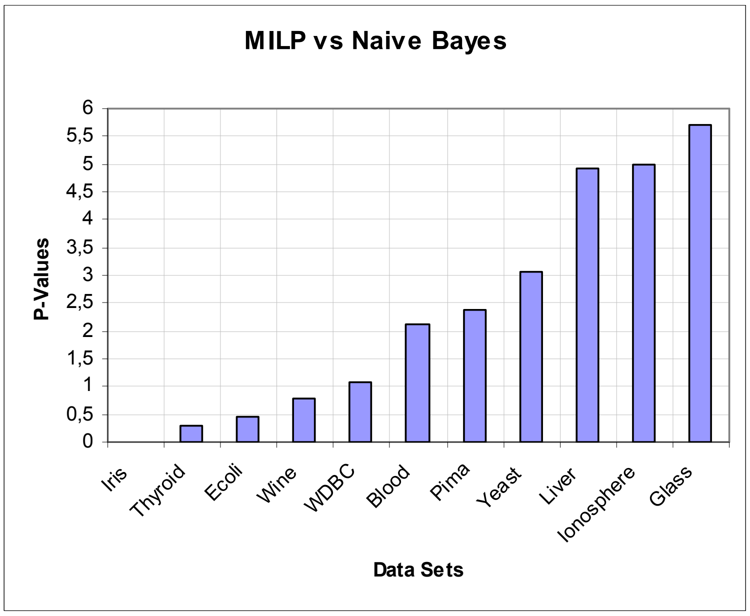

Figure 5.8 P-value graph of MILP versus Naïve Bayes for UCI Benchmark data sets.

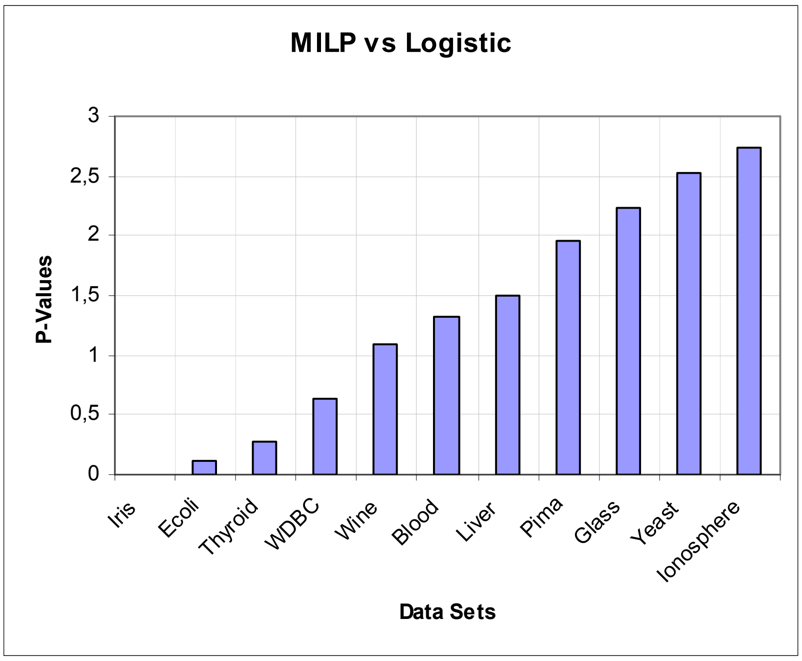

Figure 5.9 P-value graph of MILP versus Logistic for UCI Benchmark data sets.

5.6 Problematic Instance Analysis

In order to analyze whether there exists any relation between the performance of the proposed approach and the number of problematic instances in the data sets, we investigate the results of these data sets in detail. In Table 5.33, the number of problematic instances is given by average, maximum and minimum values for each of the protein folding type benchmark data sets. Furthermore, the number of problematic instances for 1189 and 25PDB data sets change from one run to another as 10FCV results are obtained for them (Figure 5.10 – Figure 5.20).

As it can be observed from the Table 5.33, the number of problematic instances does not affect the performance of the proposed approach. For example, for the data sets WDBC and thyroid which have 67% and 10% problematic instances proposed approach achieved approximately 97% accuracy. Hence, considering only the number of problematic instances could not be sufficient to analyze the difficulty of the data sets.

Table 5.33 Number of problematic instances for each of UCI repository data sets.

Data Set Name	**Accuracy (%)**	**Accuracy Rank**	**% of Av. Problematic Instances**	**Number of Problematic Instances**		
				Average	**Max.**	**Min.**
Ionosphere	94.59%	2	60%	212	217	206
Pima	81.25%	1	87%	666	677	641
Blood	79.95%	1	89%	664	666	662
WDBC	97.36%	1	67%	458	471	430
Liver	73.33%	2	80%	275	279	269
Wine	94.94%	4	6%	10	12	9
Iris	96.00%	2	15%	22	25	16
Thyroid	97.21%	1	10%	21	24	15
Glass	76.17%	2	65%	140	143	134
Ecoli	87.50%	2	48%	162	170	151
Yeast	63.00%	1	88%	1299	1307	1290

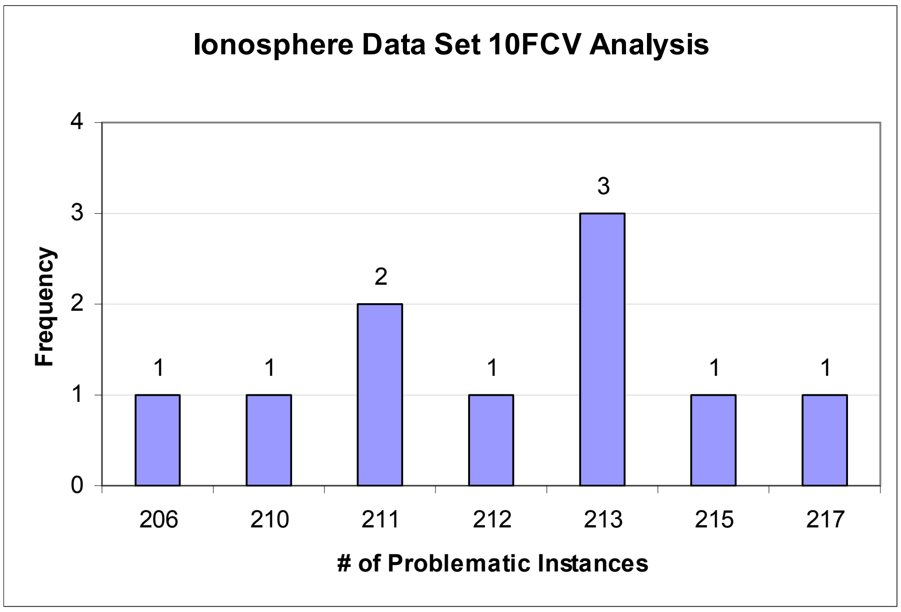

Figure 5.10 The number of problematic instances for Ionosphere data set.

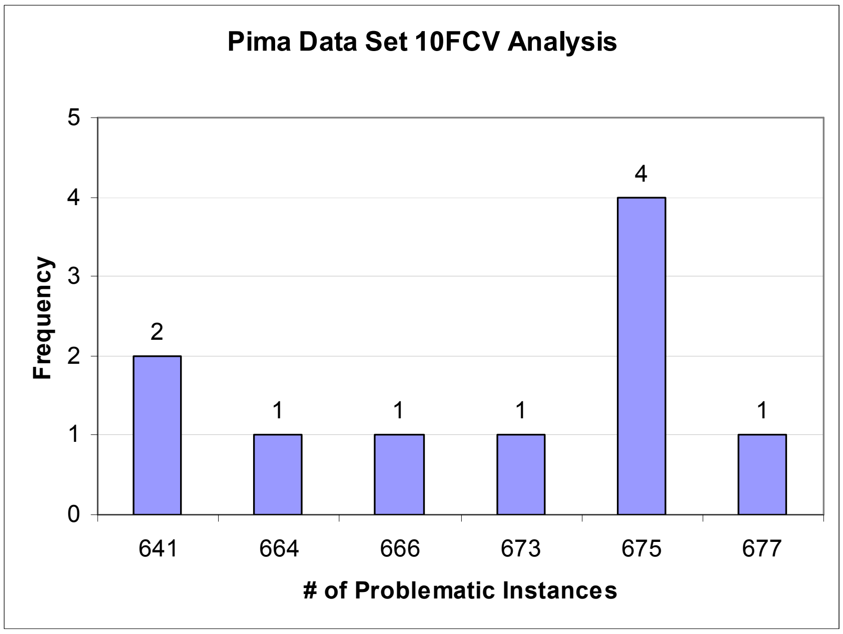

Figure 5.11 The number of problematic instances for Pima data set.

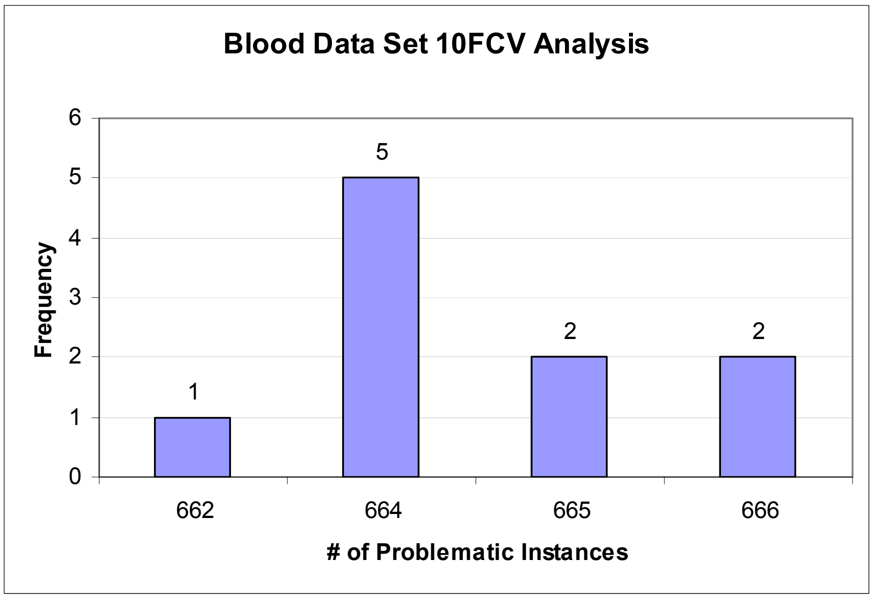

Figure 5.12 The number of problematic instances for Blood data set.

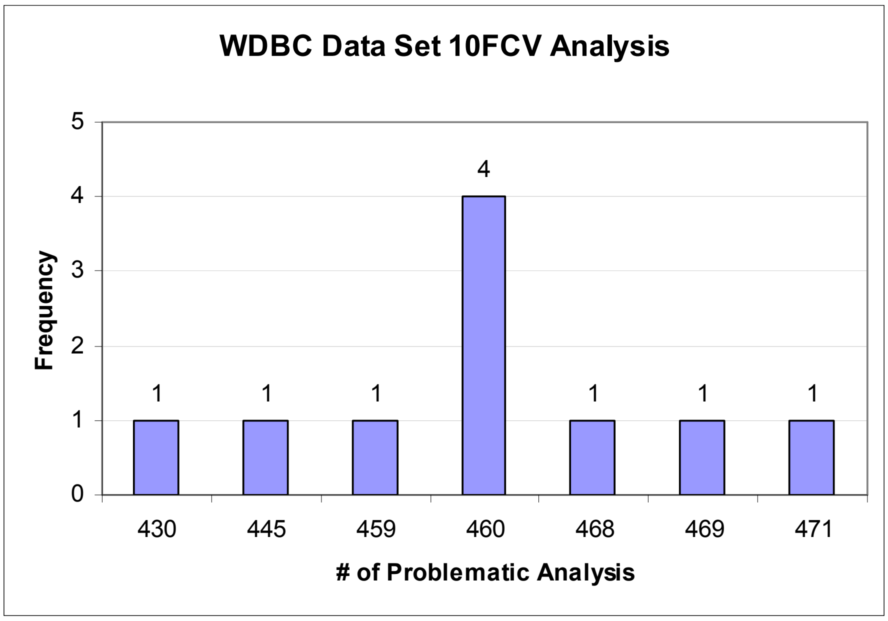

Figure 5.13 The number of problematic instances for WDBC data set.

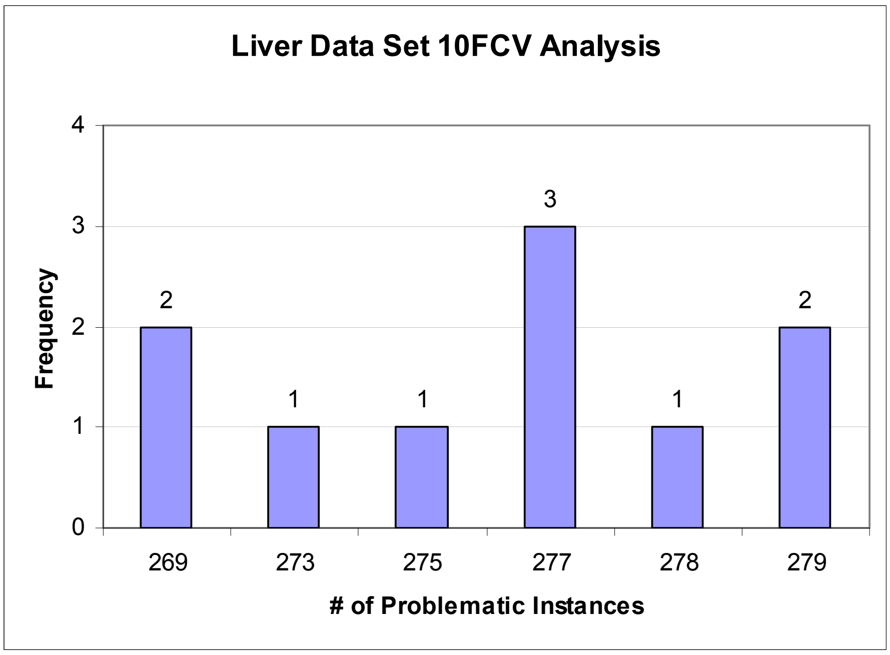

Figure 5.14 The number of problematic instances for Liver data set.

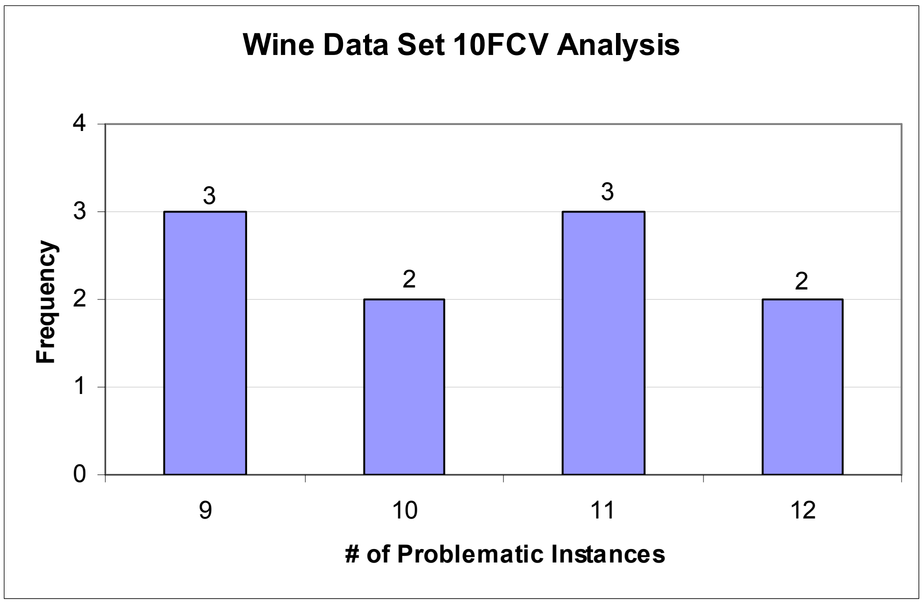

Figure 5.15 The number of problematic instances for Wine data set.

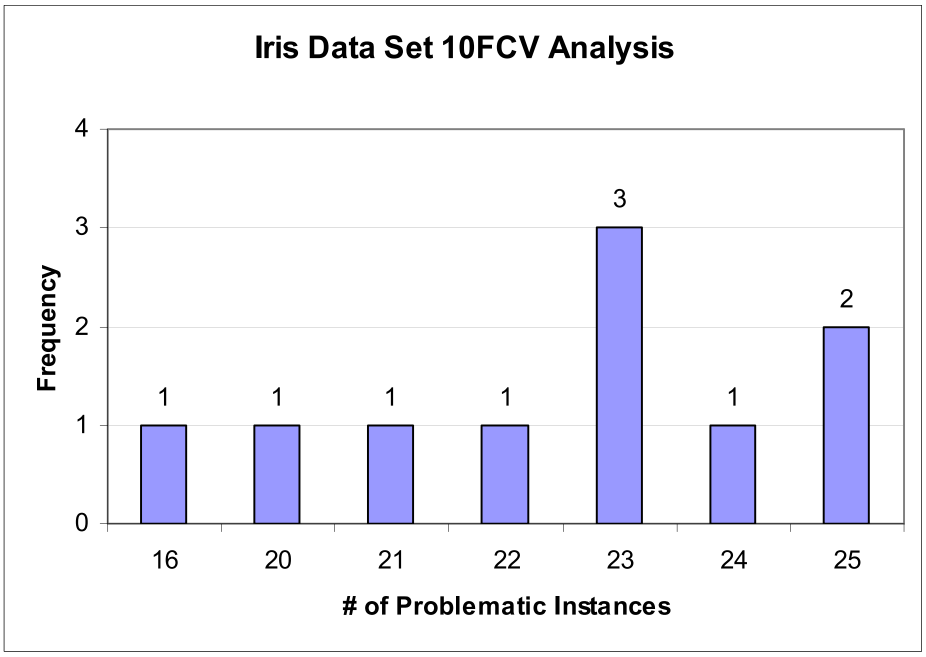

Figure 5.16 The number of problematic instances for Iris data set.

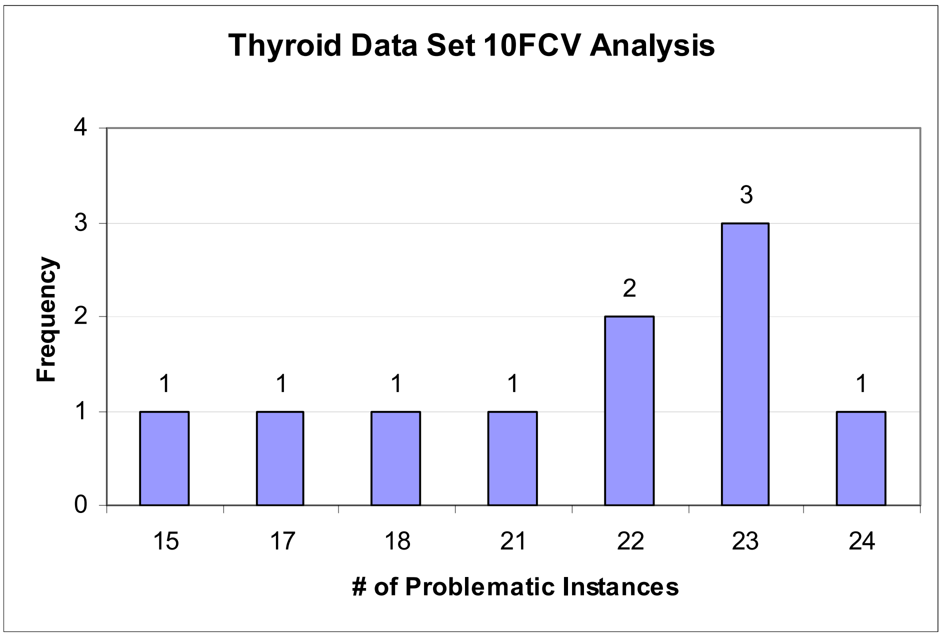

Figure 5.17 The number of problematic instances for Thyroid data set.

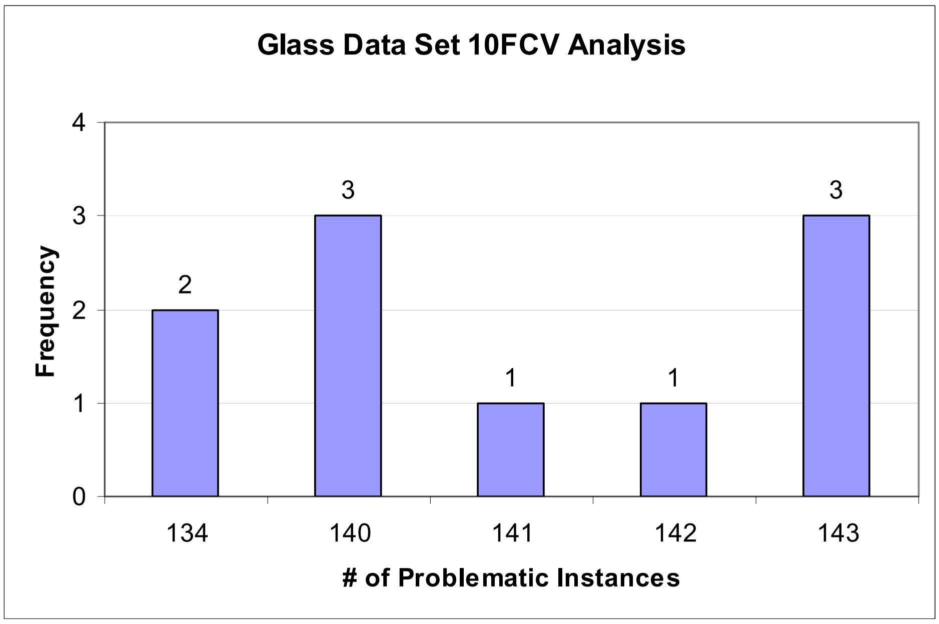

Figure 5.18 The number of problematic instances for Glass data set.

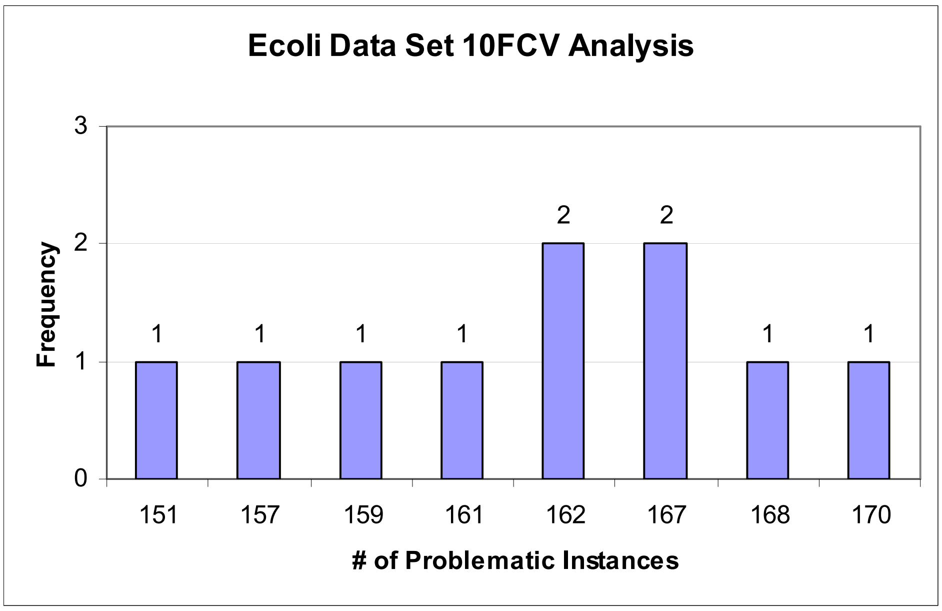

Figure 5.19 The number of problematic instances for Ecoli data set.

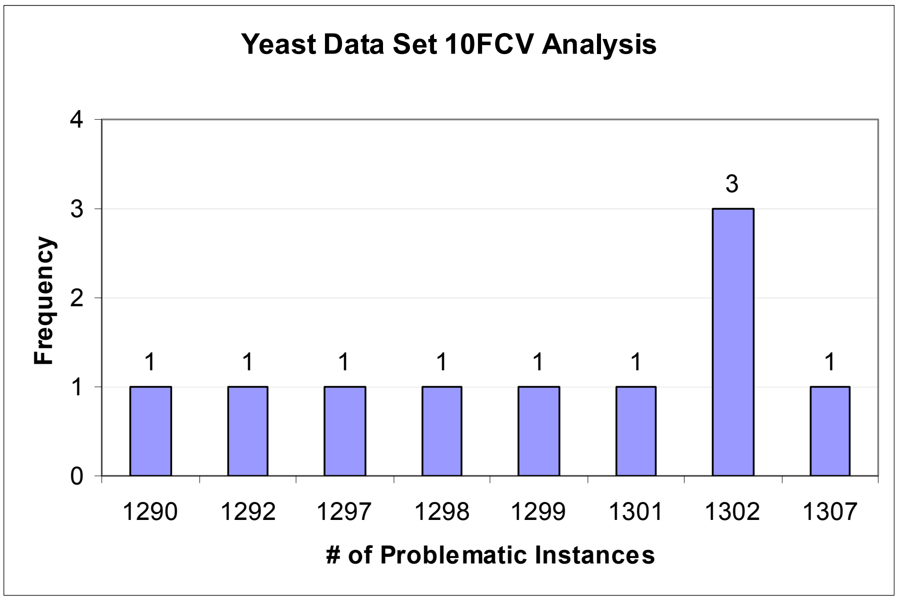

Figure 5.20 The number of problematic instances for Yeast data set.

Chapter 6

CONCLUSION

With the rapid increase in the availability of data for exploration and analysis, it is important to develop techniques that efficiently perform data mining studies. As data classification is one of the important issues in these studies, many researchers study this concept. Classification involves the supervised assignment of data points to predefined or known classes. Here, there exists a collection of classes with labels and the problem is to label a new instance as belonging to one or more of the classes. The field of data classification is wide and covers a broad range of areas including bioinformatics, decision sciences, finance, sports and health care. A large number of data classification methods have been developed to date; however, each of them has several drawbacks which make them unattractive. Thus, researchers have been studying to develop more accurate and more efficient methods or to improve the existing methods.

In this thesis, a new three-stage mathematical programming based hyper-box enclosure approach for multi-class data classification problem is proposed. A mixed-integer programming model is developed for representing existence of hyper-boxes which define the boundaries of the classes for the training set. In order to overcome the computational difficulties for large data sets, a three-stage approach is developed for training part analysis of hyper-box enclosure approach. The performance of the model is tested by the testing part of the proposed method and compared with existing multi-class data classification methods on two widely used challenging problems; the protein folding type prediction problem and UCI Repository benchmark problems.

The developed three-stage MILP based hyper-box enclosure approach to multi-class data classification is described in Chapter 3. In the training part of the proposed approach, the characteristics of data points belonging to a certain class are determined by the construction of hyper-boxes. The hyper-boxes define the boundaries of the classes that include all or some of the points in that set. In order to represent the existence of hyper-boxes and their boundaries, a mixed-integer programming model is developed.

Solving the proposed MILP formulation to optimality is computationally expensive for large multi-group data classification problems. The major source of computational difficulty is the potentially large number of binary variables. Hence, we proposed a three-stage decomposition algorithm for obtaining solutions to MILP model. Instances that are difficult to classify are identified in the first stage that is referred to as preprocessing. Moreover, sub grouping and seed finding algorithms are applied to improve the computational efficiency. With greater emphasis given to these observations, solution to the problem is obtained in the second stage using the MILP formulation. Last, final assignments, elimination of box intersections and box combination procedures are carried out in the third step.

After distinguishing characteristics of the classes are determined in the training part, the performance of the model is tested by the distance based algorithm introduced in testing problem formulation part. While the original and proposed testing algorithms are compared and investigated in detail, the advantages of proposed testing algorithm are shown. If a new data point with an unknown membership arrives, it is necessary to assign this data point to one of the classes. For each member of the test data set, testing algorithm is applied and assignments to a class are done. After all, by checking the original classes of the test set samples the performance of the developed model is evaluated.

The proposed model is illustrated on a small illustrative example. By this illustrative example, the main steps of the developed three-stage MILP based hyper-box enclosure approach are understood. Moreover, the comparison of the results of distinct models available for data classification is performed. The suggested model's result is accurate and efficient in this small example with regard to the other methods listed in Table 3.3.

In Chapter 4, proposed three-stage MILP approach is applied to the protein folding type prediction problem. Different performance evaluation techniques and measures are examined in order to investigate the details of results and compare different algorithms. The performance of proposed three-stage MILP based approach is compared with the results in [97], [100] and [23] for nine distinct data sets. Two independent datasets (225 training - 510 testing and 1601 training - 2438 testing) results are calculated and pretty good results are obtained. Furthermore, LOO test results are given for 138, 253, 277, 35 and 498 protein data sets and 10FCV results are studied for 1189 and 25PDB data sets. Results indicate that proposed MILP approach gives generally high accuracy values and mostly rank in the first or second position. Moreover, P-value analyses show that MILP approach is statistically significantly better than the existing distance based algorithms HD, ED and CC algorithm. Moreover, MILP approach is statistically better than the LibSVM and well-known WEKA classifiers for protein folding type prediction problems on given nine distinct benchmark data sets. In summary, proposed MILP based hyper-box enclosure approach is a powerful and efficient computational method for predicting folding types of proteins with its favorable results and characteristics.

In Chapter 5, the performance of proposed three-stage approach is evaluated on eleven UCI repository benchmark data sets [108]. In order to observe the performance of the proposed MILP based hyper-box enclosure approach, the eleven data sets including Ionosphere, Pima, Blood, WDBC, Liver, Wine, Iris, Thyroid, Glass, Ecoli and Yeast are tested. First five of them are binary-class data classification data sets and the rest are multi-class data sets. In order to compare the results of proposed MILP approach, WEKA classification algorithms J48, RBF Network, Logistic, Naïve Bayes (NB), SMO, Random Forest (RF) and IB1 are studied (Table 5.3). Moreover, well-known support vector machine implementation LibSVM given by [111] is also studied. Using these data sets, the proposed three-stage MILP model is solved in GAMS [120] using ILOG CPLEX Solver version 10.0 [121] on a notebook computer with Inter Pentium M 1.73 Ghz Processor and 512 MB of RAM. For each data set, 10 different runs are carried out and average 10FCV results are calculated. In order to analyze the results in detail, average sensitivity (SEN), average specificity (SPE), MCC and S values of each of the protein data sets are investigated (Table 5.16 - Table 5.28). Depending on the data set characteristics, proposed data classification approach works well for each of the classes. In order to evaluate if there is any statistical significant difference between the existing and proposed data classification approaches tested on the same data sets, *P*-value (paired test) analysis are carried out. For each of the eleven UCI Repository data sets, MILP approach generally achieves high accuracy values and mostly ranks in first three positions with respect to accuracy. Furthermore, MILP is statistically significantly better than the existing data classification methods for these benchmark problems on given eleven distinct benchmark data sets.

In conclusion, this thesis introduces a new three-stage mathematical programming based hyper-box enclosure method for multi-class data classification problem. One of the most important characteristics of the proposed approach is allowing the use of hyper-boxes for defining the boundaries of the classes that enclose all or some of the points in that set. In other words, if necessary, more than one hyper-box is constructed for a specific class in the training part. Moreover, well-construction of the boundaries of each class provides the lack of misclassifications in the training set and indirectly improves the accuracy of the model. In addition, the model does not need to know the underlying distribution of the training data set and learns from the training set in a reasonable time. With only one parameter to be initialized, the suggested model is simple and easily understandable.

Furthermore, the proposed model can be used for both binary and multi-class data classification problems without any modifications or additions. The accuracy, simplicity and understandability of the proposed model are favorable. Proposed three-stage MILP approach is applicable to obtain solutions to large multi-class data classification problems. These characteristics make the proposed approach efficient, simple and easily implementable.

The advantage of the mathematical programming approach in the context of supervised classification lies in its power to model more complex real world problems. Future studies should further evaluate the performance of the proposed approach on data sets with categorical attributes. Since the proposed approach depends on a geometrical idea, it is efficient for data sets including continuous and integer valued attributes. In literature, there exist data classification problems which include both categorical and numerical attributes. Hence, MILP approach could be modified in order to deal with categorical attributes. Moreover, overall method could be implemented in a computer package and could be parallelized. Finally, proposed data classification approach could be implemented in WEKA. In that case, there will be some solver related problems since MILP approach needs an IP solver such as GAMS. If these problems could be solved, MILP approach could be tested by many researchers by the help of WEKA.

BIBLIOGRAPHY

[1] M. S. Chen, J. Han, and P. S. Yu, Data Mining: An overview from a database perspective, IEEE Transactions Knowledge and Data Engineering, 8 (1996), 866-883.

[2] R. J. Roiger, and M. W. Geatz, Data mining-a tutorial based primer, Addison Wesley Press, (2003).

[3] H. Edelstein, Building profitable customer relationships with data mining, Two Crows Corporation, (2003).

[4] A. Jagota, Data analysis and classification for bioinformatics, Bay Press, (2000).

[5] S. M. Weiss, and C. A. Kulikowski, Computer systems that learn: classification and prediction methods from statistics, neural networks, machine learning and expert systems, Morgan Kaufmann, San Mateo, CA, (1991).

[6] A. Blum, Neural Networks in C++, An Object-Oriented Framework for building Connectionist Systems, John Wdey & Sons, New York, (1992).

[7] G. E. Hinton, Connectionist learning procedures, Artificial Intelligence, 40 (1999), 185-234.

[8] H. White, Learning in artificial neural networks: a statistical perspective, Neural Computing, 1 (1989), 425-464.

[9] V.N. Vapnik, Statistical learning theory, Wiley, New York (1998).

[10] Y. Yajima, Linear programming approaches for multi category support vector machines, European Journal of Operational Research, 162 (2) (2005), 514-531.

[11] H. Kim, S. Pang, H. Je, D. Kim, S. Y. Bang, Constructing support vector machine ensemble, Pattern Recognition, 36 (2003), 2757-2767.

[12] C. W. Hsu, and C. J. Lin, A comparison of methods for multi-class support vector machines, IEEE Transactions on Neural Networks, 13 (2) (2002), 415-425.

[13] S.S. Erenguc, G.J. Koehler, Survey of mathematical programming models and experimental results for linear discriminant analysis, Managerial and Decision Economics, 11 (1990), 215-225.

[14] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, Wadsworth International Group, Monterey, CA (1984).

[15] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan-Kaufmann, San Mateo, Ca (1993).

[16] J. Y. Lee, S Olafsson, Multi attribute decision trees and decision rules, In: Triantaphyllou, Felici (eds.), Data Mining and Knowledge Discovery Approaches Based on rule Induction Techniques, (2006), 327-358.

[17] S. Olafsson, X. Li, and S. Wu, Operations Research and Data Mining, European Journal of Operational Research, 187 (3) (2008), 1429-1448.

[18] E. Fix, J. J. Hodges, Discriminatory analysis: non-parametric discrimination: Consistency properties. Report No. 4, USAF School of Aviation Medicine, Randolph Field, TX, (1951).

[19] T. M. Cover, and P. E. Hart, Nearest Neighbor Pattern Classification, Institute of Electrical and Electronics Engineers Transactions on Information Theory, 13 (1) (1967), 21-27.

[20] A. S. Hadi, S. Chatterjee, and B. Price, Regresion Analysis by Example, John Wiley & Sons, New York, 3rd Edition, (2000).

[21] F. Pernkopf, Bayesian network classifiers versus selective *k*-NN classifer, Pattern Recognition, 38 (1) (2005), 1-10.

[22] P. N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, Boston: Pearson Education, (2006).

[23] A. Kurgan, and L. Homaeian, Prediction of structural classes for protein sequences and domains – Impact of prediction algorithms, sequence representation and homology, and test procedures on accuracy, Pattern Recognition, 39 (2006), 2323-2343.

[24] S. Jahandideh, P. Abdolmaleki, M. Jahandideh, and S. H. S. Hayatshahi, Novel hybrid method for the evaluation of parameters contributing in determination of protein structural classes, Journal of Theroretical Bilogy, 244 (2007), 275-281.

[25] J. Adem, W. Gochet, Mathematical Programming based heuristics for improving LP-generated classifiers for the multi-class supervised classification problem, European Journal of Operational Research, 168 (2006), 181-199.

[26] D. J. Hand, Discrimination and Classification, John Wiley, Chichester, (1981).

[27] A. R. Webb, Statistical Pattern Recognition, John Wiley & Sons Ltd., England, (2002).

[28] P. Lachenbruch, and R. Mickey, Estimation of error rates in discriminant analysis, Technometrics, 10 (1968), 1–11.

[29] M. G. Kendall, A. Stuart, and J. K. Ord, The advanced Theory of Statistics, Volume 3, Design and Analysis and Time Series, Chapter 44, Griffin, London, (1983).

[30] G. J. McLachlan, Discriminant Analysis and Statistical Pattern Recognition, John Wiley, New York, (1992).

[31] J. Hertz, A. Krogh, and R. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley, (1991).

[32] P. Simpson, Fuzzy min-max neural networks – Part 1: Classification, IEEE Transactions Neural Networks, 3 (5) (1992), 776-786.

[33] G. P. Zhang, Neural networks for classification: a survey, IEEE Transactions on Systems AMn and Cybernetics Part C – Applications and Reviews, 30 (4) (2000), 451-461.

[34] M. C. Bishop, Neural networks for pattern recognition, Oxford University, Press: New York, (1997).

[35] D. P. Bertsekas and J. N. Tsitsiklis, Neural Dynamic Programming, Athena Scientific, Belmont, MA, (1996).

[36] R. O. Duda, P. E. Hart, Pattern classification and scene analysis, John Wiley & Sons, New York, (1973).

[37] R. O. Duda, P. E. Hart, and D. G. Stock, Pattern classification, Second Edition, John Wiley & Sons, New York, (2001).

[38] T. Kohonen, Self organizing masps, Springer-Verlag, New York, (1995).

[39] P. A. Devijver, and J. V. Kittler, Pattern Recognition: A Statistical Approach, Prentice Hall, Englewood Cliffs, (1982).

[40] B. V. Dasarathy, J. S. Snchez, and S. Townsend, Nearest neighbor editing and condensing tools – synergy exploitation, Pattern Analysis & Applications, 3 (2000), 19-30.

[41] K. Hans, Data Mining: concepts and techniques, Morgan Kauffman, San Mateo, CA, (2001).

[42] G. Carpenter, S. Grossberg, A massively parallel architecture for a self-organizing neural pattern recognition machine, Comput. Vision Graphics Image Understanding, 37 (2) (1987), 54-115.

[43] Z. Pawlak, Rough set, International Journal Foundations of Computer Science, 11(1982), 341-356.

[44] J. Banzan, H. Nguyen, A. Skowron, J. Stepaniuk, Some logic and rough set applications for classifying objects, ICS Research Report No. 34, Warsaw University of Technology, (1994).

[45] H. Nguyen, A. Skowron, P. Synak, Discovery of data patterns with applications to decomposition and classification problems, Rough Sets in Knowledge Discovery, Physica-Verlag, Wurzburg, (1998).

[46] D. Kim, Data classification based on tolerant rough set, Pattern Recognition, 34 (2001), 1613-1624.

[47] R. Fisher, The use of multiple measurements in taxonomic problem, Ann. Eugenics, 7 (1936), 179-188.

[48] J. L. Castro, J. J. Castro-Schez, and J. M. Zurita, Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems, Fuzzy Sets and Systems, 101 (1999), 331-342.

[49] Y. C. Chen, and S. M. Chen, Constructing membership functions and generating fuzzy rules using genetic algorithms, Proceedings of the 2001 Ninth National Conference on Fuzzy Theory and Its Applications, Chungli, TAoyuan, Taiwan, Republic of China, (2001), 195-200.

[50] T. P. Hong, and C.Y. Lee, Induction of fuzzy rules and membership functions from training examples, Fuzzy Sets and Systems, 84 (1996), 33-47.

[51] H. L. Lin, and S. M. Chen, A new method for generating weighted fuzzy rules from training instances using genetic algorithms, Proceedings of the 6th Conference on Artificial Intelligence and Applications, Kaohsiung, Taiwan, Republic of China, (2001), 628-633.

[52] T. P. Wu, S. M. Chen, A new method for constructing membership functions and fuzzy rules from training examples, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 29 (1999), 25-40.

[53] S. M. Chen, and Y. D. Fang, A new approach for handling the Iris data classification problem, International Journal of Applied Science and Engineering, 1 (2005), 37-49.

[54] F. Uney, and M. Turkay, "A Mixed-Integer Programming Approach to the Multi-Class Data Classification Problem", European Journal of Operational Research, 173 (3) (2006), 910-920.

[55] S. Cheng, F. Y. Shih, An improved incremental training algorithm for support vector machines using active query, Pattern Recognition, 40 (2007), 964-971.

[56] P. Mitra, C. A. Murthy, S. K. Pal, A probabilistic active support vector learning algorithm, IEEE Trans. Pattern Anal. Mach. Intell., 26 (3) (2004), 413-418.

[57] N. A. Syed, H. Liu, K. K. Sung, Incremental learning with support vector machines, Proceedings of the International Joint Conference on Artificial Intelligent, Stockholm, Sweden, July 1999.

[58] C. Campbell, N. Cristianini, A. Smola, Query learning with large margin classifiers, Proceedings of the 17th International Conference on Machine Learning, Stanford University, CA, June 2000, pp. 111-118.

[59] J. Wetson, C. Watkins, Multi-class support vector machines, Department of Computer Science, Royal Holloway University of London Technical Report, SD2TR298204.

[60] M. L. Zhu, S. F. Chen, X. D. Liu, Sphere-structured support vector machines for multi-class pattern recognition, In: Lecture notes in computer science, 2639 (2003), 589-593.

[61] J. Wang, P. Neskovic, and L. N. Cooper, Bayes classification based on minimum bounding spheres, Neurocomputing, 70 (2007), 801-808.

[62] D. Lee, J. Lee, Domain described support vector classifier for multi-classification problems, Pattern Recognition, 40 (2007), 41-51.

[63] K. P. Bennett, Decision tree construction via linear programming, In: Proceedings of the 1994 International Conference on Very Large Data Bases, (1994), 487-499.

[64] H. Kennedy, C. Chinniah, P. Bradbeer, L. Morss, The construction and evaluation of decision tress: a comparison of evolutionary and concept learning methods, In: Come D., Shapiro, J. (Eds.), Evolutionary Computing Lecture Notes in Computer Science, Springer, Berlin, (1997), pp. 147-161.

[65] A. Niimi, E. Tazaki, Genetic programming combined with association rule algorithm for decision tree construction, In: Fourth International Conference on Knowledge-based Intelligent Engineering Systems and Allied Technologies, Brighton, UK, (2000), pp. 746-749.

[66] W. V. Gehrlein, General mathematical programming formulations for the statistical classification problem, Operation Research Letters, 5 (1986), 299-304.

[67] N. Freed, F. Glover, A linear programming approach to the discriminant problems, Decision Sciences, 12 (1981), 68-74.

[68] O. L. Mangasarian, Linear and nonlinear separation of patterns by linear programming, Operations Research, 13 (1965), 444-452.

[69] D. J. Hand, Discrimination and classification, Wiley: Chichester, (1981).

[70] N. Freed, F. Glover, Simple but powerful goal programming models for discriminant problems, European Journal of Operational Research, 7 (1981), 44-60.

[71] W. Gochet, A.Stam, V. Srinivasan, and S. Chen, multi-group discriminant analysis using linear programming, Operations Research, 45 (1997), 213-225.

[72] S. M. Bajgier, and A. V. Hill, An experimental comparison of statistical and linear programming approaches to the discriminant problem, Decision Sciences, 13 (1982), 604-618.

[73] J. M. Wilson, Integer programming formulations of statistical classification problems, Omega, 24 (1996), 681-688.

[74] A. Stam, E. A. Joachimsthaler, A comparison of a robust mixed-integer approach to existing methods for establishing classification rules for the discriminant problem, European Journal of Operational Research, 46 (1990), 113-122.

[75] G. J. Koehler, Considerations for mathematical programming models in discriminant analysis, Managerial and Decision Economics, 11 (1990), 227-234.

[76] J. J. Glen, Integer programming methods for normalization and variable selection in mathematical programming discriminant analysis models, Journal of Operational Research Society, 50 (1990), 1043-1053.

[77] R. Pavur, C. Loucopulos, Examining optimal criterion weights in mixed integer programming approaches to the multiple-group classification problem, Journal of Operational Research Society, 46 (1995), 626-640.

[78] K. F. Lam, J. W. Moy, Improved linear programming formulations for the multi-group discriminant problem, Journal of Operational Research Society, 47 (1996), 1526-1529.

[79] A. P. D. Silva, A. Stam, A mixed integer programming algorithm for minimizing the training sample misclassification cost in two-group classification, Annals of Operations Research, 74 (1997), 129-157.

[80] J. J. Glen, An iterative mixed integer programming method for classification accuracy maximizing discriminant analysis, Computers & Operations Research, 30 (2003), 181-198.

[81] J. W. Chinneck, Fast heuristics for the maximum feasible subsystem problem, INFORMS Journal on Computing, 13 (3) (2001), 210-223.

[82] P. A. Rubin, Heuristic solution procedures for a mixed-integer programming discriminant model, Managerial and Decision Economics, 11 (1990), 255-266.

[83] D. Tax, R. Duin, Using two-class classifiers for multiclass classification, Proceedings 16th International Conference on Pattern Recognition, Quebec City, Canada, vol. II, IEEE Computer Society Press, Los Alamitos, 2002, pp. 124-127.

[84] A. Stam, C. T. Ragsdale, On the classification gap in mathematical-programming-based approaches to the discriminant problem, Naval Research Logistics, 39 (1992), 545-559.

[85] J. J. Glen, A comparison of standard and two-stage mathematical programming discriminant analysis methods, European Journal of Operational Research, 171 (2006), 496-515.

[86] M. Gromiha, and S. Selvaraj, Protein secondary structure prediction in different structural classes, Protein Engineering, 11 (1998), 249–251.

[87] F. Uney Yuksektepe, O. Yilmaz, and M. Turkay, Prediction of secondary structures of proteins using a two-stage method, Computers and Chemical Engineering, 32 (2008), 78–88.

[88] K. C. Chou, and C. T. Zhang, Prediction of protein structural classes, Crit. Rev. Biochem. Mol. Biol., 30 (1995), 275-349.

[89] I. Bahar, A. R. Atilgan, R. L. Jernigan, B. Erman, Understanding the recognition of protein structural classes by amino acid composition, Proeins, 29 (1997), 172-185.

[90] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, The Protein Data Bank, Nucleic Acids Research, 28 (2000), 235-242.

[91] A. Murzin, S. Brenner, T. Hubbard and C. Chotia, SCOP: a structural classification of protein database for the investigation of sequence and structures, J. Mol. Biol., 247 (1995), 536-540.

[92] H. Nakashima, K. Nishikawa, and T. Ooi, The folding type of a protein is relevant to the amino acid composition, J. Biochem., 99 (1986), 152-162.

[93] K. C. Chou, A novel approach to predicting protein structural classes in a (10-1)-D amino acid composition space, Proteins, 21 (1995), 319-344.

[94] W. Kabsch, C. Sander, Dictionary of protein secondary structures: pattern recognition of hydrogen-bonded and geometrical features, Bioploymers, 22 (1983), 2577-2637.

[95] F. Eisenhaber, C. Frommel, and P. Argos, Prediction of secondary structural content of proteins from their amino acid composition alone. II. The paradox with secondary structural class PROTEINS: Structure, Function, and Genetics, 25 (1996), 169-179.

[96] Z.-X. Wang, Z. Yuan, How good is the prediction of protein structural class by the component-coupled method?, Proteins, 38 (2000), 165-175.

[97] K. C. Chou, G. M. Maggiora, Domain Structural class prediction, Protein Engineering, 11 (1998), 523-538.

[98] A. Andreeva, D. Howorth, S. Brenner, T. Hubbard, C. Chothia, A. Murzin, SCOP database in 2004: refinements integrate structure and sequence family data, Nucleic Acid Res., 32 (2004), 226-229.

[99] K. C. Chou, Does the folding type of a protein depend on its amino acid composition?, FEBS Letters, 363 (1995), 127-131.

[100] G. P. Zhou, An intriguing controversy over protein structural class prediction Journal of Protein Chemistry, 17 (1998), 729-738.

[101] K. C. Chou, W. M. Liu, G. M. Maggiora, and C. T. Zhang, Prediction and classification of domain structural classes, PROTEINS: Structure, Function, and Genetics, 31 (1998), 97-103.

[102] Y. D. Cai, Is it a paradox or misinterpretation?, PROTEINS: Structure, Function, and Genetics, 43 (2001), 336-338.

[103] W. S. Bu, Z. P. Feng, Z. Zhang, and C. T. Zhang, Prediction of protein (domain) structural classes based on amino-acid index, Eur. J. Biochem., 266 (1999), 1043-1049.

[104] Y. D. Cai, and G. P. Zhou, Prediction of protein structural classes by neural network, Biochimie., 82 (2000), 783-785.

[105] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, Networks Flows: Theory, Algorithms and Applications, Prentice Hall, New Jersey, (1993).

[106] F. Uney, A mixed-integer programming approach to multi-class data classification problem, Msc. Thesis, Koç University, September 2005.

[107] I. H. Witten, and E. Frank, Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, (2005).

[108] A. Asuncion, and D. J. Newman, UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html], Irvine, CA: University of California, School of Information and Computer Science, (2007).

[109] Y. D. Cai, X. J. Liu, X. B. Xu, and G. P. Zhou, Support vector machines for predicting protein structural class, BMC Bioinformatics, 2 (3) (2001).

[110] Y. D. Cai, X. J. Liu, X. B. Xu and K. C. Chou, Support vector machines for prediction of protein structural class, Journal of Theoretical Biology, 221 (2003), 115-120.

[111] C. C. Chang, and C. J. Lin, LIBSVM: a library for support vector machines, 2001, Software [http://www.csie.ntu.edu.tw/~cjlin/libsvm]

[112] G.H. John, P. Langley P, Estimating continuous distributions in Bayesian classifiers, Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Mateo, 1995, 338–345.

[113] A. Saha, C.L. Wu, D.S. Tang, Approximation, approximation dimension reduction and nonconvex optimization using linear superpositions of gaussians, IEEE Trans. Comput., 42 (1993), 1222–1233.

[114] D. Aha, D. Kibler, Instance-based learning algorithms, Mach. Learn., 6 (1991), 37–66.

[115] R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, 1993.

[116] L. Breiman, Random forests, Mach. Learn., 45 (1) (2001), 5–32.

[117] W. Cohen, Fast effective rule induction, Proceeding of the 12th International Conference on Machine Learning, Lake Tahoe, CA, 1995, 115–123.

[118] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R. Murthy, Improvements to Platt's SMO algorithm for SVM classifier design, Neural Comput., 13 (3) (2001), 637–649.

[119] S. le Cessie, J.C. van Houwelingen, Ridge estimators in logistic regression, Appl. Stat., 41 (1) (1992), 191–201.

[120] A. Brooke, D. Kendrick, A. Meeraus, R. Raman, GAMS: A User's Guide, GAMS Development Co., Washington, DC, 1998.

[121] Ilog, 2006. ILOG CPLEX 10.0 User's Manual.

APPENDIX A: PROTEIN FOLDING TYPE PREDICTION DATA SETS

Table A.1 The 138 Protein Domains.

36 all-α domains					
1hbiA|W.C.	1sctA|W.C.	1ytc_|W.C.	1boc_|W.C.	1ctz_|W.C.	1troA|W.C.
1fipA|W.C.	1hddC|W.C.	1dprA|65-136	1tnt_|W.C.	1erc_|W.C.	2tct_|W.C.
1aca_|W.C.	1vasA|W.C.	1lynA|W.C.	1hsm_|W.C.	1rprA|W.C.	3wrp_|W.C.
1pou_|W.C.	1arqA|W.C.	1mykA|W.C.	1mylA|W.C.	1bpd_|9-91	1lis_|W.C.
1olhA|W.C.	1pesA|W.C.	1rpo_|W.C.	1hns_|W.C.	1tag_|57-177	1rhgA|W.C.
1tyc_|228-319	1oxy_|1-379	1pgn_|177-473	1csi_|W.C.	1phb_|W.C.	1lla_|2-379
29 all-β domains					
1mdtA|381-535	1cgt_|580-684	1gcs_|1-85	1pnf_|1-140	1png_|5-140	1gog_|151-537
1gog_|1-150	1tnfA|W.C.	1hivA|W.C.	2ctvA|W.C.	1apnA|W.C.	1cgt_|383-494
1bib_|271-317	1bfb_|W.C.	2bfh_|W.C.	1bfg_|W.C.	1arc_|W.C.	1hpcA|W.C.
1bcmA|481-560	1hvc_|W.C.	1hbp_|W.C.	1fen_|W.C.	1slfB|W.C.	1kraC|2-129
1azm_|W.C.	1srgA|W.C.	1sleB|W.C.	1cyhA|W.C.	3cysA|W.C.	
32 α/β domains					
1cgt_|1-382	1cxe_|1-382	1btb_|W.C.	1brsD|W.C.	1fnd_|155-314	1garA|W.C.
4ts1A|1-217	1selA|W.C.	1cdoA|176-324	1hldA|175-324	1horA|W.C.	3pgk_|W.C.
1cia_|W.C.	1pnt_|W.C.	2hnp_|W.C.	1tho_|W.C.	1lam_|1-159	1olcA|W.C.
1gdtA|1-140	3hsc_|3-188	1idm_|W.C.	1cde_|W.C.	1cddA|W.C.	1pkm_|396-530
1mhtA|W.C.	1alhA|W.C.	8atcA|1-150	2ctc_|W.C.	1dr1_|W.C.	2rslA|W.C.
1drj_|W.C.	2bgt_|W.C.				
41 α+β domains					
1fut_|W.C.	2baa_|W.C.	1aec_|W.C.	2rat_|W.C.	2rns_|W.C.	1mrk_|W.C.
1rbd_|W.C.	1kraA|W.C.	1pgb_|W.C.	2igg_|W.C.	2secI|W.C.	1mldA|145-313
3monA|W.C.	1frtA|1-178	1fkj_|W.C.	2tecI|W.C.	1lttA|W.C.	1ltaA|W.C.
3mdsA|93-203	1egpA|W.C.	1mns_|3-132	1grl_|137-190	1r1dS	2act_|W.C.
1comA|W.C.	1sphA|W.C.	1gaeO|149-312	1mstA|W.C.	1grb_|364-478	1molA|W.C.
1lklA|W.C.	1lckA|117-226	1sceA|W.C.	1tsy_|W.C.	3b5c_|W.C.	1xrb_|1-101
1tbpA|61-155	1xrc_|1-101	1glv_|123-316	3dni_|W.C.	1dnkA|W.C.	

* Each domain is represented by a symbol of X|Y, where first four character of X is the corresponding PDB code and the fifth character indicates the specific chain of the protein. If it is _, then the corresponding protein has only one chain. If Y=W.C., it means the domain is constituted by the whole chain. Otherwise, Y contains two number to indicate starting and end points along the sequence.

Table A.2 The 253 Protein Domains.

63 all-α domains					
1hbiA|W.C.	1sctA|W.C.	1ytc_|W.C.	1crj_|W.C.	1hddC|W.C.	1glm_|W.C.
1dprA|65-136	1tnt_|W.C.	1erc_|W.C.	1aca_|W.C.	1vasA|W.C.	2tct_|W.C.
1lynA|W.C.	1hsm_|W.C.	1rprA|W.C.	1rpo_|W.C.	1pou_|W.C.	2ts1_|228-319
1cdn_|W.C.	1arqA|W.C.	1mykA|W.C.	1mylA|W.C.	1bpd_|9-91	1csh_|W.C.
1olhA|W.C.	1pesA|W.C.	1hns_|W.C.	1tag_|57-177	4ts1A|228-319	1oelA|2-136
1tyc_|228-319	1oxy_|1-379	1pgn_|177-473	1csi_|W.C.	1phb_|W.C.	1llp_|W.C.
1troA|W.C.	3wrp_|W.C.	3sdhA|W.C.	1ycc_|W.C.	1enh_|W.C.	1phc_|W.C.
1dtr_|65-191	1tns_|W.C.	1bal_|W.C.	1erl_|W.C.	2abd_|W.C.	1rtm1|73-104
2end_|W.C.	1lis_|W.C.	1aab_|W.C.	1rhgA|W.C.	1ropA|W.C.	1lla_|2-379
1octC|5-75	4icb_|W.C.	1parA|W.C.	2bpfA|9-91	1olgA|W.C.	1fiaA|W.C.
1hnr_|W.C.	2wrpR|W.C.	2pgd_|177-473			
58 all-β domains					
1mdtA|381-535	1cgt_580-684	1gcs_|1-85	1pnf_|1-140	1pnf_|5-140	2sil_|W.C.
1gog_|1-150	1tnfA|W.C.	2ctvA|W.C.	1apnA|W.C.	1bib_|271-317	2pec_|W.C.
1bfb_|W.C.	2bfh_|W.C.	1bfg_|W.C.	1arc_|W.C.	1bcmA|481-560	1gtrA|339-547
1hpxA|W.C.	1hvc_|W.C.	1hbp_|W.C.	1fen_|W.C.	1slfB|W.C.	1gof_|151-537
1srgA|W.C.	1sleB|W.C.	1cyhA|W.C.	3cysA|W.C.	1gog_|151-537	1htp_|W.C.
1cgt_|383-494	1hug_|W.C.	1hpcA|W.C.	1kraC|2-129	1ddt_|381-535	2kauC|2-129
1cdg_582-686	1aac_|W.C.	4gcr_|1-85	1pgs_|4-140	1gof_|1-150	1hcb_|W.C.
1tnrA|W.C.	1thw_|W.C.	1scs_|W.C.	1bglA|731-1023	1bia_|271-317	1kapP|247-470
1ltsD|W.C.	4fgf_|W.C.	1fnb_|19-154	1arb_|W.C.	1bco_|481-560	2cpl_|W.C.
1difA|W.C.	1hbq_|W.C.	1cdg_|383-495	1sriA|W.C.		
61 α/β domains					
1cgt_|1-382	1cxe_|1-382	1btb_|W.C.	1brsD|W.C.	1fnd_|155-314	7acn_|2-528
4ts1A|1-217	1cdoA|176-324	1hldA|175-324	1horA|W.C.	2secE|W.C.	1ctt_|1-150
1cia_|W.C.	1pnt_|W.C.	2hnp_|W.C.	1trx_|W.C.	1lam_|1-159	3pgk_|W.C.
1gdtA|1-140	3hsc_|3-188	1cde_|W.C.	1cddA|W.C.	1mhtA|W.C.	1aliA|W.C.
2ctc_|W.C.	1dr1_|W.C.	2anhA|W.C.	1xab_|W.C.	1raiA|1-150	1xaa_|W.C.
2bgt_|W.C.	1drk_|W.C.	1olcA|W.C.	1cdg_|1-382	1bta_|W.C.	2bgu_|W.C.
1fnb_|155-314	2ts1_|1-217	2ohxA|175-324	1deaA|W.C.	1cseE|W.C.	1ubsB|W.C.
3cla_|W.C.	1phr_|W.C.	2hnq_|W.C.	2trxA|W.C.	1trkA|535-680	2dri_|W.C.
1pkm_|396-530	1lcpA|1-159	2rslA|W.C.	1hpm_|4-188	1garA|W.C.	1ora_|1-149
1hmy_|W.C.	7aatA|W.C.	1ulb_|W.C.	1ack_|W.C.	2ctb_|W.C.	2olbA|W.C.
8dfr_|W.C.					
71 α+β domains					
1fut_|W.C.	2baa_|W.C.	1aec_|W.C.	2rat_|W.C.	2rns_|W.C.	1puc_|W.C.
1rbd_|W.C.	1kraA|W.C.	1pgb_|W.C.	2igg_|W.C.	3monA|W.C.	1xrb_|1-101
1frtA|1-178	1fkj_|W.C.	2secI|W.C.	1egpA|W.C.	2tecI|W.C.	1ltsA|W.C.
3mdsA|93-203	1mns_|3-132	1gr1_|137-190	1rldS|W.C.	1comA|W.C.	1sryA|111-421
1gaeO|149-312	1mstA|W.C.	1grb_|364-478	1lklA|W.C.	1lckA|177-226	2glt_|123-316
1sphA|W.C.	1sceA|W.C.	1tsy_|W.C.	3b5c_|W.C.	1tbpA|61-155	1tlcA|W.C.
1xrc_|1-101	1glv_|123-316	3dni_|W.C.	1dnkA|W.C.	1mrk_|W.C.	2dnjA|W.C.
1ltaA|W.C.	1lttA|W.C.	1fus_|W.C.	1cnsA|W.C.	2act_|W.C.	1cyo_|W.C.
7rsa_|W.C.	2kauA|W.C.	1igd_|W.C.	3cox_|319-450	1molA|W.C.	1mldA|145-313
1fruA|1-178	1fkd_|W.C.	1cseI|W.C.	1mngA|93-203	1vih_|W.C.	1ytbA|61-155
2mnr_|3-132	1oelA|137-190	3rubS|W.C.	2chsA|W.C.	1gadO|149-312	1mrj_|W.C.
3sicI|W.C.	2ms2A|W.C.	3grs_|364-478	1lkkA|W.C.	1hid_|W.C.	

Table A.3 The 359 Protein Domains.

82 all-α domains					
1hbiA|W.C.	1sctA|W.C.	1ytc_|W.C.	1boc_|W.C.	1ctz_|W.C.	2ts1_|228-319
1fipA|W.C.	1hddC|W.C.	1dprA|65-136	1tnt_|W.C.	1bbl_|W.C.	1csh_|W.C.
1erc_|W.C.	1aca_|W.C.	1vasA|W.C.	1lynA|W.C.	1hme_|W.C.	1oelA|2-136
1hsm_|W.C.	1gnc_|W.C.	1rprA|W.C.	1pou_|W.C.	1cdn_|W.C.	1llp_|W.C.
1cih_|W.C.	1arqA|W.C.	1mykA|W.C.	1mylA|W.C.	1bpd_|9-91	1phc_|W.C.
1olhA|W.C.	1pesA|W.C.	1rpo_|W.C.	1hns_|W.C.	1tag_|57-177	1rtm1|73-104
1bod_|W.C.	2pccB|W.C.	4ts1A|228-319	1tyc_|228-319	1lgaA|W.C.	1lla_|2-379
1oxy_|1-379	1nol_|1-379	1pgn_|177-473	1yeb_|W.C.	2utgA|W.C.	1fiaA|W.C.
3gly_|W.C.	1csi_|W.C.	1csc_|W.C.	1phb_|W.C.	3fisA|W.C.	2pgd_|177-473
1troA|W.C.	3wrp_|W.C.	1trrA|W.C.	1grl_|6-136	1raq_|W.C.	2wrpR|W.C.
1afb1|73-104	3sdhA|W.C.	1ycc_|W.C.	1enh_|W.C.	1dtr_|65-191	1glm_|W.C.
1tns_|W.C.	1bal_|W.C.	1erl_|W.C.	2abd_|W.C.	2end_|W.C.	2tct_|W.C.
1lis_|W.C.	1aab_|W.C.	1rhgA|W.C.	1ropA|W.C.	1octC|5-75	1hnr_|W.C.
4icb_|W.C.	1parA|W.C.	2bpfA|9-91	1olgA|W.C.		
85 all-β domains					
1mdtA|381-535	1cgt_|580-684	1cxe_|582-686	1aaj_|W.C.	1mdaA|W.C.	1sriA|W.C.
1gcs_|1-85	1pnf_|1-140	1png_|5-140	1gog_|1-150	1tnfA|W.C.	1hcb_|W.C.
1hivA|W.C.	1thu_|W.C.	2ctvA|W.C.	2tunA|W.C.	1apnA|W.C.	2cpl_|W.C.
2cna_|W.C.	1bib_|271-317	1ltaD|W.C.	1bfb_|W.C.	2bfh_|W.C.	1kapP|247-470
1bfg_|W.C.	1bas_|W.C.	1fnd_|19-154	1arc_|W.C.	1bcmA|481-560	2sil_|W.C.
1hpxA|W.C.	1thv_|W.C.	1hshA|W.C.	1bzm_|W.C.	1cpiA|W.C.	2pec_|W.C.
1hvc_|W.C.	1hefE|W.C.	1hvsA|W.C.	1gtsA|339-547	1hbp_|W.C.	1gof_|151-537
1fen_|W.C.	1fga_|W.C.	1erb_|W.C.	1slfB|W.C.	1azm_|W.C.	1htp_|W.C.
1srgA|W.C.	1srjA|W.C.	1ptsA|W.C.	1sleB|W.C.	1cyhA|W.C.	1cdg_|383-495
3cysA|W.C.	2sim_|W.C.	1gog_|151-537	1cgt_|383-494	1cxe_|383-495	2kauC|2-129
1hug_|W.C.	1mikA|W.C.	1huh_|W.C.	1akl_|247-470	1hpcA|W.C.	1hbq_|W.C.
1kraC|2-129	1ddt_|381-535	1cdg_|582-686	1aac_|W.C.	4gcr_|1-85	1gtrA|339-547
1pgs_|4-140	1gof_|1-150	1tnrA|W.C.	1thw_|W.C.	1scs_|W.C.	1difA|W.C.
1bglA|731-1023	1bia_|271-317	1ltsD|W.C.	4fgf_|W.C.	1fnb_|19-154	1bco_|481-560
1arb_|W.C.					
99 α/β domains					
1cgt_|1-382	1cxe_|1-382	1cgv_|1-382	1btb_|W.C.	1brsD|W.C.	1ulb_|W.C.
1cxf_|1-382	1fnd_|155-314	4ts1A|1-217	1selA|W.C.	1cdoA|176-324	1xaa_|W.C.
1hldA|175-324	1horA|W.C.	2secE|W.C.	1cia_|W.C.	1frn_|155-314	2bgu_|W.C.
1pnt_|W.C.	2hnp_|W.C.	1tybE|1-217	1tho_|W.C.	1tkbA|535-680	1ack_|W.C.
1lam_|1-159	1bllE|1-159	1gdtA|1-140	3hsc_|3-188	1idm_|W.C.	1ubsB|W.C.
1ngi_|4-188	1atr_|2-188	1cde_|W.C.	1grcA|W.C.	1cddA|W.C.	2dri_|W.C.
1mhtA|W.C.	1ama_|W.C.	1alhA|W.C.	1ula_|W.C.	1ngb_|4-188	2ctb_|W.C.
1rhd_|1-149	1trx_|W.C.	1amn_|W.C.	8atcA|1-150	1acj_|W.C.	1ora_|1-149
1alkA|W.C.	2ctc_|W.C.	1dr1_|W.C.	1drj_|W.C.	1hqaA|W.C.	2olbA|W.C.
1ajdA|W.C.	1acl_|W.C.	1ngg_|3-188	1ajcA|W.C.	1dbp_|W.C.	8dfr_|W.C.
1xab_|W.C.	1raiA|W.C.	1scnE|W.C.	1ttqB|W.C.	1wsyB|W.C.	7acn_|2-528
1orb_|1-149	1ajaA|W.C.	2anhA|W.C.	5acn_|1-528	5cpa_|W.C.	1ctt_|1-150
2bgt_|W.C.	1drk_|W.C.	1acmA|1-150	1ngh_|4-188	1olcA|W.C.	1aliA|W.C.
1ctu_|1-150	1cdg_|1-382	1bta_|W.C.	1fnb_|155-314	2ts1_|1-217	3pgk_|W.C.
2ohxA|175-324	1deaA|W.C.	1cseE|W.C.	3cla_|W.C.	1phr_|W.C.	7aatA|W.C.
2hnq_|W.C.	2trxA|W.C.	1trkA|535-680	1pkm_|396-530	1lcpA|1-159	1hmy_|W.C.

2rs1A|W.C.	1hpm_|4-188	1garA|W.C.			
93 α+β domains					
1fut_|W.C.	2baa_|W.C.	1aec_|W.C.	2rat|W.C.	2rns_|W.C.	1lkkA|W.C.
1ras_|W.C.	1ssbA|W.C.	1rbd_|W.C.	1kraA|W.C.	1pgx_|W.C.	1cyo_|W.C.
1pgb_|W.C.	1igcA|W.C.	2igg_|W.C.	2igh_|W.C.	2secI|W.C.	1mldA|145-313
1coy_|319-450	3monA|W.C.	1frtA|1-178	1fkj_|W.C.	2tecI|W.C.	1hid_|W.C.
1lttA|W.C.	1egl_|W.C.	1sbnI|W.C.	3mdsA|93-203	1vig_|W.C.	1ytbA|61-155
1egpA|W.C.	1fkl_|W.C.	1mns_|3-132	1grl_|137-190	1fccC|W.C.	1mrj_|W.C.
1rldS|W.C.	1comA|W.C.	1sphA|W.C.	1gaeO|149-312	1mstA|W.C.	1puc_|W.C.
1grb_|364-478	1lklA|W.C.	1lcjA|W.C.	1lckA|117-226	1sceA|W.C.	1xrb_|1-101
1setA|111-421	1sibI|W.C.	1tsdA|W.C.	1htlA|W.C.	1bmsA|W.C.	1ltsA|W.C.
2hpr_|W.C.	1tsy_|W.C.	1tys_|W.C.	3b5c_|W.C.	1tbpA|61-155	1sryA|111-421
1xrc_|1-101	1glv_|123-316	2tscA|W.C.	3dni_|W.C.	1dnkA|W.C.	2glt_|123-316
4mdhA|155-333	1mrk_|W.C.	1ltaA|W.C.	1ltgA|W.C.	1fus_|W.C.	1tlcA|W.C.
1cnsA|W.C.	2act_|W.C.	7rsa_|W.C.	2kauA|W.C.	1igd_|W.C.	2dnjA|W.C.
3cox_|319-450	1molA|W.C.	1fruA|1-178	1fkd_|W.C.	1cseI|W.C.	3grs_|364-478
1mngA|93-203	1vih_|W.C.	2mnr_|3-132	1oelA|137-190	3rubS|W.C.	2ms2A|W.C.
2chsA|W.C.	1gadO|149-312	3sicI|W.C.			

Table A.4 The 1601 Protein Domains.

273 all-α domains					
3sdhA|W.C.	1flp_|W.C.	2hbg_|W.C.	1bvc_|W.C.	2myc_|W.C.	1utg_|W.C.
2mb5_|W.C.	1mls_|W.C.	1mbw_|W.C.	1mod_|W.C.	2mga_|W.C.	5cscA|W.C.
1mba_|W.C.	1mbs_|W.C.	1mygA|W.C.	1ymb_|W.C.	1mniA|W.C.	1oxa_|W.C.
1emy_|W.C.	1lht_|W.C.	1myt_|W.C.	1eca_|W.C.	2gdm_|W.C.	1aorA|211-605
1lh1_|W.C.	2hhbA|W.C.	2hbcA|W.C.	1cohA|W.C.	1dshA|W.C.	1oelA|2-136
2mhbA|W.C.	1hdsA|W.C.	1hdaA|W.C.	2pghA|W.C.	1pbxA|W.C.	1pshA|W.C.
2mhbB|W.C.	1hbcB|W.C.	1cohB|W.C.	2hhe3|W.C.	1fdhG|W.C.	4p2p_|W.C.
2hhbB|W.C.	1hdsB|W.C.	1hdaB|W.C.	2pgh3|W.C.	1pbxB|W.C.	2ztaA|W.C.
2lhb_|W.C.	1ithA|W.C.	1ash_|W.C.	1hlb_|W.C.	1cpcA|W.C.	1ifk_|W.C.
1grj_|2-79	1sryA|1-110	1idsA|W.C.	3sdpA|5-834	1isaA|1-82	1ccd_|W.C.
1abmA|1-83	1mngA|1-92	1ycc_|W.C.	1csw_|W.C.	1csv_|W.C.	2cts_|W.C.
1hrc_|W.C.	1ccr_|W.C.	5cytR|W.C.	1cyc_|W.C.	3c2c_|W.C.	1cpt_|W.C.
1c2rA|W.C.	1cxc_|W.C.	1cry_|W.C.	1cot_|W.C.	1cc5_|W.C.	1bvp1|1-120
1cor_|W.C.	451c_|W.C.	2mtaC|W.C.	1cyi_|W.C.	1fcdC|W.C.	1ecmA|W.C.
1enh_|W.C.	1yrnA|W.C.	1lfb_|W.C.	1octC|102-161	1ftt_|W.C.	1pp2R|W.C.
1hdp_|W.C.	1ocp_|W.C.	1hom_|W.C.	1ftz_|W.C.	1hcrA|W.C.	1bunA|W.C.
1gdtA|141-183	1mbe_|W.C.	1pdnC|W.C.	1bia_|1-63	1lea_|W.C.	1d66A|49-64
1cgpA|138-205	1hstA|W.C.	1ghc_|W.C.	1fliA|W.C.	1etc_|W.C.	1ifl_|W.C.
1stwA|W.C.	1hks_|W.C.	2hts_|W.C.	1dtr_|4-64	1dtr_|65-191	1g1m_|W.C.
1tns_|W.C.	2spcA|W.C.	1fc2C|W.C.	1bal_|W.C.	2pdd_|W.C.	1phc_|W.C.
1erl_|W.C.	1erd_|W.C.	1erp_|W.C.	1acp_|W.C.	2abd_|W.C.	1fiaA|W.C.
2end_|W.C.	1lis_|W.C.	1aab_|W.C.	1hma_|W.C.	1hryA|W.C.	2sblB|150-838
1bfmA|W.C.	1mmoG|W.C.	1lpe_|W.C.	1le4_|W.C.	1le2_|W.C.	1csmA|W.C.
2asr_|W.C.	2ligA|W.C.	256bA|W.C.	2ccyA|W.C.	1bbhA|W.C.	1ppa_|W.C.
1cgn_|W.C.	1cgo_|W.C.	2hmzA|W.C.	2mhr_|W.C.	2tmvP|W.C.	1clpA|W.C.
1cgmE|W.C.	1bucA|233-383	3mddA|242-395	1bcfA|W.C.	1fha_|W.C.	1pyiA|72-117
1hrs_|W.C.	1rcd_|W.C.	1ribA|W.C.	1mmo3|W.C.	1rhgA|W.C.	2ifo_|W.C.
1bgc_|W.C.	1bgeA|W.C.	1lki_|W.C.	3hhrA|W.C.	1ilk_|W.C.	1clc_|135-574
1gmfA|W.C.	1rcb_|W.C.	1itl_|W.C.	1hulA|W.C.	1ir1_|W.C.	7cpp_|W.C.
1rfbA|W.C.	1ropA|W.C.	1eciA|W.C.	1octC|5-75	1lmb3|W.C.	1prcC|W.C.
1r69_|W.C.	2cro_|W.C.	1adr_|W.C.	1neq_|W.C.	1pnrA|3-58	2tct_|W.C.
1lccA|W.C.	1coo_|W.C.	1mdyA|W.C.	4icb_|W.C.	1cb1_|W.C.	1poc_|W.C.
1sra_|W.C.	1rro_|W.C.	1cdp_|W.C.	1pvb_|W.C.	5pa1_|W.C.	1bbc_|W.C.
1rtp1|W.C.	1top_|W.C.	5tnc_|W.C.	1rec_|W.C.	2scpA|W.C.	1rtm1|73-104
2sas_|W.C.	1cll_|W.C.	1lin_|W.C.	3cln_|W.C.	1cfd_|W.C.	1ifj_|W.C.
1osa_|W.C.	1scmB|W.C.	1scmC|W.C.	1parA|W.C.	1mntA|W.C.	1csh_|W.C.
1cmbA|W.C.	1dsbA|65-128	2gstA|85-217	1glqA|79-209	1gsrA|77-207	2hpdA|W.C.
1gssA|77-207	1hna_|85-217	1gseA|81-222	2gsq_|76-202	1gta_|81-218	2wrpR|W.C.
1bmtA|651-740	1c5a_|W.C.	1hyp_|W.C.	1lpt_|W.C.	1lip_|W.C.	1fps_|W.C.
1bip_|W.C.	2bpfA|9-91	1olgA|W.C.	1sakA|W.C.	1hnr_|W.C.	1poa_|W.C.
1hueA|W.C.	1aep_|W.C.	1axn_|W.C.	1ala_|W.C.	1hvd_|W.C.	4bp2_|W.C.
2ran_|W.C.	1ann_|W.C.	1tadA|57-177	1gia_|51-181	1ezm_|154-298	1hup_|88-111
8tlnE|156-316	4tmnE|156-316	1npc_|157-317	2ts1_|228-319	2hmx_|W.C.	1ifm_|W.C.
1llp_|W.C.	1aru_|W.C.	2cyp_|W.C.	1ccc_|W.C.	1cpd_|W.C.	2pgd_|177-473
1mnp_|W.C.	1apxA|W.C.	1mhlA|W.C.	1mypA|W.C.	1pth_|74-583	1hc2_|5-398
2abk_|W.C.	1gln_|306-468	1lla_|2-379			

461 all-β domains

1bec_|3-117	8fabA|3-105	7fabL|1-103	1bafL|1-108	1bbdL|1-114	1r081|W.C.
1bbjL|1-109	1hilA|1-108	1dbaL|1-107	1dfbL|1-106	1igfL|1-107	1cov1|W.C.
1igiL|1-107	1igmL|W.C.	1indL|2-109	2f19L|1-108	2fb4L|1-109	1dhx_|W.C.
2fbjL|1-109	1fgvL|W.C.	2imm_|W.C.	1fvcA|W.C.	1ggbL|1-107	1hplA|337-449
1acyL|1-108	1mamL|1-108	1nbvL|1-112	1tetL|1-107	1flrL|1-112	1bvp1|121-254
6fabL|1-108	1gigL|1-110	2cgrL|1-112	1figL|1-108	1frgL|1-108	1thw_|W.C.
1vfaA|W.C.	1jhlL|W.C.	3hf1L|1-106	3hfmL|1-108	1jelL|1-108	1lte_|W.C.
1ncaL|1-108	1forL|1-108	1eapA|1-107	1mrdL|1-108	1fbiL|1-107	2ayh_|W.C.
1rmfL|1-112	1fptL|1-108	1ikfL|1-107	1lmkA|2-127	1igcL|1-108	1celA|W.C.
1ibgL|2-107	1mlbA|1-108	1nmbL|W.C.	1opgL|1-107	1nsnL|1-107	1oacA|301-724
1iaiL|1-108	1iaiM|1-109	1plgL|1-112	1ivlA|W.C.	1reiA|W.C.	1pht_|W.C.
2rhe_|W.C.	1bjmA|W.C.	1wtlA|W.C.	1breA|W.C.	1mcoL|1-111	1gbrA|W.C.
1mcdA|1-111	1mceA|1-111	1mcwM|1-111	3cd4_|1-97	1cid_|1-105	1qweA|W.C.
1hnf_|4-104	1cdcA|W.C.	1cd8_|W.C.	1bec_|118-246	8fabA|106-208	1qorA|2-135
7fabL|104-204	1bafL|109-214	1bbdL|115-219	1bbjL|110-211	1hilA|109-211	1prtD|W.C.
1dbaL|108-211	1dfbL|106-212	1igfL|108-214	1igiL|108-213	1indL|110-212	1tssA|1-93
2f19L|109-214	2fb4L|110-214	2fbjL|108-213	2fgwL|109-214	1mcpL|115-219	1pyp_|W.C.
1fvdA|109-214	1ggbL|108-211	1acyL|109-211	1mamL|109-214	1mfbL|112-212	1bgh_|W.C.
1nbvL|113-219	1tetL|108-211	1flrL|113-219	6fabL|109-214	1gigL|111-210	4fgf_|W.C.
2cgrL|113-219	1figL|108-214	1frgL|112-217	1fdlL|108-214	3hflL|107-212	2aaiB|1-135
3hfmL|109-214	1jelL|109-212	1ncaL|109-214	1forL|108-210	1eapA|108-214	1fnb_|19-154
1mrdL|109-211	1fbiL|108-214	1rmfL|113-219	1fptL|108-213	1ikfL|108-214	1eft_|313-405
1igcL|109-213	1ibgL|108-214	1mlbA|109-214	1opgL|108-214	1nsnL|108-211	1gbdA|W.C.
1iaiL|109-214	1iaiM|110-215	1plgL|113-215	1mcoL|112-216	1mcdA|112-216	1ppcE|W.C.
1mceA|112-216	1mcwM|112-216	1fc1A|238-341	1frtC|239-341	1pfc_|W.C.	1brbE|W.C.
1fruA|179-269	1bmg_|W.C.	2clrA|182-275	1hsaA|182-276	1hsbA|182-270	2gmt_|W.C.
1vabA|182-274	1hocA|182-272	1mhcA|182-272	1dlhA|82-182	1vcaA|1-90	7estE|W.C.
3cd4_|98-178	1cid_|106-177	1hnf_|105-182	1hngA|101-176	1cgx_|496-581	1nrpL|W.C.
1tlk_|W.C.	1tnm_|W.C.	1gof_538-639	1cdg_|496-581	1clc_|35-134	1ton_|W.C.
1cyg_|492-574	1ciu_|496-578	1lla_|380-628	1hc2_|399-653	1ten_|W.C.	1difA|W.C.
1ctn_|24-132	1ggtA|8-190	2hft_|1-106	1fna_|W.C.	2mcm_|W.C.	1idaA|W.C.
1cfb_|610-709	3hhrB|32-130	1ggtA|516-627	1nciA|W.C.	1spdA|W.C.	1er8E|W.C.
1noa_|W.C.	1acx_|W.C.	1akp_|W.C.	1sxcA|W.C.	1ddt_|381-535	1psoE|W.C.
1xsoA|W.C.	1srdA|W.C.	1jcv_|W.C.	1rsy_|W.C.	1cgx_|582-686	1lybA|W.C.
1exg_|W.C.	1tupA|W.C.	1ctm_|1-167	1cdg_|582-686	2pcdA|W.C.	1dro_|W.C.
1cyg_|575-680	1ciu_|579-683	1ttaA|W.C.	1ttcA|W.C.	1plc_|W.C.	1pkyA|70-157
2pcdM|W.C.	1hoe_|W.C.	2ait_|W.C.	1aac_|W.C.	1pmy_|W.C.	1epbA|W.C.
9pcy_|W.C.	1pla_|W.C.	2plt_|W.C.	1paz_|W.C.	1cyx_|W.C.	1mdc_|W.C.
1azcA|W.C.	1arn_|W.C.	1ilsA|W.C.	1azrA|W.C.	1gff1|W.C.	1pmpA|W.C.
1nif_|8-166	1afnA|11-166	1aozA|1-129	2bpa1|W.C.	2tbvA|W.C.	2cpl_|W.C.
2stv_|W.C.	1smvA|W.C.	1bmv1|W.C.	4sbvA|W.C.	4rhv1|W.C.	1fbl_|272-466
1cwpA|W.C.	2bbvA|W.C.	1bbt1|W.C.	2cas_|W.C.	1vcaA|91-199	1nscA|W.C.
1cgx_|383-495	1ppi_|404-496	2cba_|W.C.	1heb_|W.C.	1vmoA|W.C.	2pec_|W.C.
1cqpA|9-137	1ctm_|231-249	2kauC|2-129	1ruj1|W.C.	1tme1|W.C.	4gcr_|1-85
1lpbB|337-449	1hgiA|W.C.	1scs_|W.C.	1loeA|W.C.	1cpn_|W.C.	1xnb_|W.C.
1bia_|271-317	2pni_|W.C.	1semA|W.C.	1psf_|W.C.	1ltsD|W.C.	1prtF|W.C.
1se2_|1-120	1ino_|W.C.	1gpc_|W.C.	1barA|W.C.	1abrB|1-140	2cnd_|11-124
1arb_|W.C.	1gbeA|W.C.	2tgt_|W.C.	1trnA|W.C.	4gch_|W.C.	1elt_|W.C.
1ahtL|W.C.	1hcgA|W.C.	1hvlA|W.C.	2rspA|W.C.	4er4E|W.C.	1htrP|W.C.
4cms_|W.C.	1dynA|W.C.	1hbq_|W.C.	1mup_|W.C.	1ftpA|W.C.	1sriA|W.C.
2rmcA|W.C.	2sil_|W.C.	1gof_|151-537	1cyg_|379-491	1hny_|404-496	1cnx_|W.C.

1cva_|W.C.	1dlc_|290-499	1tsp_|W.C.	1wapA|W.C.	1gpr_|W.C.	1pov0|W.C.
2rhn1|W.C.	2bb2_|2-85	1pgs_|4-140	4hmgA|W.C.	2ltnA|W.C.	1lgcA|W.C.
1sltA|W.C.	1xnd_|W.C.	1ckaA|W.C.	1shg_|W.C.	1hsq_|W.C.	2ohxA|1-174
1chpD|W.C.	1snc_|W.C.	1asyA|68-204	2prd_|W.C.	1rip_|W.C.	2afgA|W.C.
1tie_|W.C.	1ndh_|3-125	2sga_|W.C.	1hpgA|W.C.	1gbt_|W.C.	1bit_|W.C.
1ppfE|W.C.	1ihsL|W.C.	1hylA|W.C.	1lmwA|W.C.	4hvpA|W.C.	1mvpA|W.C.
1ppmE|W.C.	1mpp_|W.C.	1bw3_|W.C.	1pls_|W.C.	1rbp_|W.C.	1hms_|W.C.
1cbs_|W.C.	1stsB|W.C.	1cynA|W.C.	1nnc_|W.C.	2bbkH|W.C.	1ciu|383-495
1amg_|358-416	4ca2_|W.C.	1cnb_|W.C.	1msaA|W.C.	1lxa_|W.C.	1htp_|W.C.
1f3g_|W.C.	1pvc1|W.C.	1r1a1|W.C.	1prr_|1-90	1gof_|1-150	1knb_|W.C.
1lesA|W.C.	1sba_|W.C.	1hlcA|W.C.	1xyn_|W.C.	1shfA|W.C.	1lckA|63-116
1srl_|W.C.	6adhA|1-174	1bovA|W.C.	1sty_|W.C.	1krs_|W.C.	1mjc_|W.C.
1prcH|37-258	1i1b_|W.C.	1wbc_|W.C.	2pia_1-103	1sgpE|W.C.	1sgt_|W.C.
1tabE|W.C.	1try_|W.C.	1elg_|W.C.	1hahL|W.C.	3rp2A|W.C.	2snv_|W.C.
1hteA|W.C.	1sivA|W.C.	2apr_|W.C.	1hrnA|W.C.	1gtrA|339-547	1pkn_|116-217
1rlbE|W.C.	1ifc_|W.C.	1cbiA|W.C.	2aviA|W.C.	1clh_|W.C.	1nnb_|W.C.
3aahA|W.C.	2aaa_|374-476	1amy_|347-402	1cim_|W.C.	1dmxA|W.C.	1kapP|247-470
2phlA|W.C.	1lab_|W.C.	2kauB|W.C.	2mev1|W.C.	1fpv_|W.C.	2sblB|7-149
1dlc_|500-643	1tnrA|W.C.	1led_|W.C.	1gbg_|W.C.	1sacA|W.C.	1xyoA|W.C.
1aboA|W.C.	1griA|1-63	1cskA|W.C.	1dehA|1-174	1prtB|88-197	1sye_|W.C.
1lylA|14-153	1csp_|W.C.	1pcrH|36-250	1ilr1|W.C.	1hce_|W.C.	1eft_|213-312
2alp_|W.C.	4ptp_|W.C.	1mctA|W.C.	3gctA|W.C.	1eleE|W.C.	1abjL|W.C.
2pkaA|W.C.	1bco_|481-560	1fivA|W.C.	1epnE|W.C.	3psg_|W.C.	1smrA|W.C.
1btn_|W.C.	1pkm_|116-217	1bbpA|W.C.	1lib_|W.C.	1opaA|W.C.	1smpI|W.C.
1hxn_|W.C.	6nn9_|W.C.	1cdg_|383-495	6taa_|374-476	1hcb_|W.C.	1hec_|W.C.
3bcl_|W.C.	1sat_|247-470	1cauA|W.C.	1bncA|331-446	1dupA|W.C.	

332 α/β domains

1cdg_|1-382	1cgx_|1-382	1cyg_|1-378	1ciu_|1-382	2aaa_|1-353	1opr_|W.C.
6taa_|1-353	1ppi|1-403	1hny_|1-403	1amg_|1-357	1amy_|1-346	1admA|W.C.
1byb_|W.C.	1ceo_|W.C.	2exo_|W.C.	1ghsA|W.C.	1ghr_|W.C.	1art_|W.C.
1xyzA|W.C.	1cbg_|W.C.	1pbgA|W.C.	1nar_|W.C.	1cnv_|W.C.	2dkb_|W.C.
1llo_|W.C.	2ebn_|W.C.	1edt_|W.C.	1ctn_|133-441	1add_|W.C.	1ack_|W.C.
2kauC|130-422	1pta_|W.C.	1nal1|W.C.	1ald_|W.C.	1fbaA|W.C.	2had_|W.C.
2acs_|W.C.	1ral_|W.C.	4enl_|142-436	1pdz_|140-433	2mnr_|133-359	1tib_|W.C.
2chr_|127-370	1oyb_|W.C.	1gox_|W.C.	2tmdA|1-340	1ltdA|98-511	1hplA|1-336
1ubsA|W.C.	1pii_|1-252	1pkm_|12-115	1pkn_|12-115	1pkyA|1-69	8dfr_|W.C.
1dik_|510-874	3rubL|148-467	1ausL|148-463	1rblA|148-475	5ru bA|138-457	1ajbA|W.C.
1tph1|W.C.	1htiA|W.C.	7timA|W.C.	1treA|W.C.	1tmhA|W.C.	4at1A|1-150
1btmA|W.C.	6xia_|W.C.	1dxiA|W.C.	2gyiA|W.C.	2xis_|W.C.	1aco_|2-528
1xih_|W.C.	4xiaA|W.C.	1xlbA|W.C.	1ximA|W.C.	2xinA|W.C.	1minA|W.C.
1brlA|W.C.	1nfp_|W.C.	1fvpA|W.C.	1tml_|W.C.	2tmdA|490-645	1agx_|W.C.
3cox_|5-318	1pbe_|1-173	1doc_|1-173	1gal_|3-324	3grs_|18-165	1abe_|W.C.
1gerA|3-146	1tde_|1-118	1npx_|1-119	2tprA|1-168	3ladA|1-158	1tlfA|W.C.
1fcdA|1-114	1dik_|377-505	7acn_|529-754	1aco_|529-754	1oelA|191-375	1lst_|W.C.
1bta_|W.C.	1bnh_|W.C.	1iceA|W.C.	1udh_|W.C.	1mla_|3-127	1lct_|W.C.
3chy_|W.C.	2chf_|W.C.	1ntr_|W.C.	1scuA|122-288	1scuB|245-388	1pxtA|28-293
2fcr_|W.C.	2fx2_|W.C.	1rcf_|W.C.	1ofv_|W.C.	4fxn_|W.C.	1oroA|W.C.
1bmtA|741-896	1ordA|1-107	1cus_|W.C.	1esc_|W.C.	2nacA|1-147	1dctA|W.C.
1gdhA|2-100	1psdA|7-107	1dldA|1-103	1fnb_|155-314	2cnd_|125-270	1ase_|W.C.
1ndh_|126-272	2pia_|104-223	2ts1_|1-217	1gtrA|8-338	1gln_|1-305	1ordA|108-569
1gpmA|208-404	2tmdA|341-489	2ohxA|175-324	6adhA|175-324	1dehA|175-324	1fssA|W.C.
1qorA|136-265	1hdcA|W.C.	1dhr_|W.C.	1hdr_|W.C.	1eny_|W.C.	1thtA|W.C.

1gadO|0-148	1gd1O|0-148	1cerO|1-148	1hdgO|1-148	1ggaO|1-164	1thg_|W.C.
1gypA|1-165	1gpdG|1-148	3gpdR|1-150	1dpgA|1-181	1dih_|2-130	1lbpB|1-336
2nacA|148-335	1gdhA|101-291	1psdA|108-295	2dldA|104-300	1m1dA|1-144	1dhfA|W.C.
2cmd_|1-145	1bmdA|0-154	1hlpA|21-146	1hyhA|21-166	9ldtA|1-162	1xaa_|W.C.
2ldx_|1-159	1ldm_|1-160	1ldnA|15-162	1llc_|13-164	1lldA|7-149	1ragA|1-150
2pgd_|1-176	1scuA|1-121	1bncA|1-114	2dln_|1-96	2glt_|1-122	3pmgA|1-190
1pydA|2-181	1pvdA|2-181	1powA|183-365	1nbaA|W.C.	1deaA|W.C.	2bgu_|W.C.
1powA|9-182	1trkA|3-337	1gky_|W.C.	1ukz_|W.C.	3adk_|W.C.	3pga1|W.C.
2ak3A|W.C.	1akeA|W.C.	1aky_|W.C.	5p21_|W.C.	1crr_|W.C.	2gbp_|W.C.
1plk_|W.C.	1tadA|27-56	1gia_|34-60	1hurA|W.C.	1eft_|1-212	2lbp_|W.C.
1dts_|W.C.	1adeA|W.C.	1nipA|W.C.	2reb_|3-268	1chd_|W.C.	1sbp_|W.C.
1cseE|W.C.	1thm_|W.C.	1st3_|W.C.	1sup_|W.C.	2sbt_|W.C.	1ovb_|W.C.
2prk_|W.C.	1meeA|W.C.	1mpt_|W.C.	3c1a_|W.C.	1qca_|W.C.	1ctt_|1-150
1eaf_|W.C.	1phr_|W.C.	2hnq_|W.C.	1yts_|W.C.	2trxA|W.C.	1lfaA|W.C.
1thx_|W.C.	3trx_|W.C.	1aazA|W.C.	1dsbA|1-64	1gp1A|W.C.	7aatA|W.C.
2gstA|1-84	1glqA|1-78	1gsrA|1-76	1gssA|1-76	1hna_|1-84	1spa_|W.C.
1gseA|2-80	2gsq_|1-75	1gta_|1-80	1trkA|535-680	1pkm_|396-530	1ulb_|W.C.
1pkn_|396-530	1pkyA|351-470	1lcpA|1-159	1eriA|W.C.	1rvaA|W.C.	1mahA|W.C.
1bam_|W.C.	1pvuA|W.C.	2rslA|W.C.	1hpm_|4-188	1ngh_|4-188	1tca_|W.C.
2btfA|2-146	2yhx_|2-202	1hkg_|2-202	1glaG|4-253	1chmA|2-156	1crl_|W.C.
2rn2_|W.C.	1gob_|W.C.	1ril_|W.C.	1vrtA|430-539	1hnvA|430-556	2ctb_|W.C.
1rdd_|W.C.	1vsd_|W.C.	1itg_|W.C.	1bco_|258-480	1kfd_|324-518	1dyr_|W.C.
1hjrA|W.C.	3pgm_|W.C.	1rpa_|W.C.	1gph1|235-465	1hmpA|W.C.	1xac_|W.C.
1ubsB|W.C.	3pgk_|W.C.	1gpb_|W.C.	1pfkA|W.C.	1gca_|W.C.	1pnrA|59-340
1mpb_|W.C.	1ovt_|5-334	1garA|W.C.	1akbA|W.C.	1aam_|W.C.	1pbn_|W.C.
1whtA|W.C.	3tgl_|W.C.	1cleA|W.C.	1lcpA|160-484	4dfrA|W.C.	1idc_|W.C.
1ora_|W.C.	1php_|W.C.	1pygA|W.C.	3pfk_|W.C.	1pea_|W.C.	2olbA|W.C.
1hslA|W.C.	1tfd_|W.C.	1hmy_|W.C.	2cstA|W.C.	1tp1A|W.C.	1gpmA|3-207
1ysc_|W.C.	1tia_|W.C.	1tahB|W.C.	1amp_|W.C.	1aliA|W.C.	1idf_|W.C.
7acn_|2-528	1mioA|W.C.	3ecaA|W.C.	2dri_|W.C.	2liv_|W.C.	1pda_|3-219
1dppA|W.C.	1lfg_|1-334				

297 α+β domains

1fus_|W.C.	9rnt_|W.C.	1rgk_|W.C.	1trpA|W.C.	1gmpA|W.C.	1ltdA|10-97
1brnL|W.C.	1bscA|W.C.	1banA|W.C.	1rms_|W.C.	1cnsA|W.C.	2polA|1-122
193l_|W.C.	1rcmA|W.C.	3lym_|W.C.	6lyz_|W.C.	1lze_|W.C.	1scuB|1-244
135l_|W.C.	1hhl_|W.C.	1ghlA|W.C.	1bqlY|W.C.	2ihl_|W.C.	1pnkA|W.C.
1lzr_|W.C.	1lz5_|W.C.	1lhk_|W.C.	2eq1_|W.C.	1lmq_|W.C.	1hlpA|147-328
1alc_|W.C.	1hml_|W.C.	4lzm_|W.C.	1l92_|W.C.	130l_|W.C.	1ldnA|163-330
163l_|W.C.	113l_|W.C.	1l24_|W.C.	1l63_|W.C.	1lyg_|W.C.	1abrA|W.C.
1l49_|W.C.	146l_|W.C.	1l15_|W.C.	1l41_|W.C.	1l01_|W.C.	1prtA|W.C.
1l98_|W.C.	1l51_|W.C.	1l71_|W.C.	1l53_|W.C.	1l95_|W.C.	1afa1|105-226
205l_|W.C.	176lA|W.C.	189l_|W.C.	153l_|W.C.	1gbs_|W.C.	1mat_|W.C.
2act_|W.C.	1ppn_|W.C.	5pad_|W.C.	1ppo_|W.C.	1hucA|W.C.	1plq_|1-126
1theA|W.C.	1gecE|W.C.	1gcb_|W.C.	1ggtA|191-515	7rsa_|W.C.	1dik_|2-376
8rat_|W.C.	1rnnE|W.C.	1rbn_|W.C.	1rbh_|W.C.	1onc_|W.C.	1pyaA|W.C.
1bsrA|W.C.	1ang_|W.C.	1agi_|W.C.	2kauA|W.C.	1napA|W.C.	1hyhA|167-329
3il8_|W.C.	1plfA|W.C.	1rhpA|W.C.	1mgsA|W.C.	1humA|W.C.	1llc_|165-333
1rtoA|W.C.	1sso_|W.C.	1sap_|W.C.	1pkp_|78-147	1igd_|W.C.	1apa_|W.C.
2ptl_|W.C.	1ubi_|W.C.	1frd_|W.C.	4fxc_|W.C.	1fxiA|W.C.	1dmaA|W.C.
1dox_|W.C.	1frrA|W.C.	2pia_|224-320	1put_|W.C.	1tssA|94-194	1prtB|4-87
1sc2_|121-239	1tif_|W.C.	3cox_|319-450	1pbe_|174-275	1doc_|174-275	1ytbA|61-155
1gal_|518-582	1molA|W.C.	1cyv_|W.C.	1stfI|W.C.	1oacA|91-185	2glt_|123-316

1std_|W.C.	1udiI|W.C.	1fruA|1-178	1dlhA|3-81	2clrA|1-181	2dnjA|W.C.
1hsaA|1-181	1hsbA|1-181	1vabA|1-181	1hocA|1-181	1mhcA|1-181	1mldA|145-313
1aak_|W.C.	2uce_|W.C.	1fkd_|W.C.	1fkr_|W.C.	1yat_|W.C.	9ldtA|163-331
1ctn_|516-560	1grj_|80-157	1dhy_|1-132	1han_|2-132	1cseI|W.C.	1lldA|150-319
1sibI|W.C.	2sniI|W.C.	1tin_|W.C.	1mngA|93-203	1abmA|84-198	1rtc_|W.C.
3sdpA|84-190	1isaA|83-192	1idsA|86-199	1ctf_|W.C.	2reb_|269-328	1esl_|1-118
1stu_|W.C.	1pkp_|4-77	1pda_|220-306	1vih_|W.C.	1gpmA|405-525	3pmgA|421-561
2mnr_|3-132	4enla_|1-141	1pdz_|1-139	2chr_|1-126	1oelA|137-190	2dln_|97-306
1fxd_|W.C.	1fdx_|W.C.	1fca_|W.C.	1clf_|W.C.	5fd1_|W.C.	1aorA|1-210
1frj_|W.C.	1fxrA|W.C.	2fxb_|W.C.	4at1B|8-100	1ragE|1-100	2cmd_|146-312
1pba_|W.C.	1spbP|W.C.	1pil_|W.C.	1nueA|W.C.	1npk_|W.C.	2ldx_|160-331
1nsqA|W.C.	1nhkR|W.C.	1urnA|W.C.	1sxl_|W.C.	2bopA|W.C.	1mrj_|W.C.
3rubL|22-147	1ausL|20-147	1rb1A|9-147	5rubA|2-137	1aps_|W.C.	1ltsA|W.C.
1ris_|W.C.	1regX|W.C.	1psdA|327-410	1mla_|128-197	1vhh_|W.C.	1hup_|112-228
1tig_|W.C.	1ife_|W.C.	1kptA|W.C.	3rubS|W.C.	1ausS|W.C.	1xrb_|1-101
1rb1M|W.C.	1dchA|W.C.	1xxaA|W.C.	2chsA|W.C.	1otfA|W.C.	1bncA|115-330
1otgA|W.C.	1gadO|149-312	1gd1O|149-312	1cerO|149-312	1hdgO|149-312	1gph1|1-234
1ggaO|165-333	1gypA|166-334	1gpdG|149-312	3gpdR|151-314	1dih_|131-240	1bmdA|155-332
1dpgA|182-412	1oacA|5-90	3sicI|W.C.	2ms2A|W.C.	1frsA|W.C.	1ldm_|161-329
3grs_|364-478	1gerA|336-450	1npx_|322-447	2tprA|358-482	3ladA|349-472	1mrg_|W.C.
1fcdA|328-401	1ezm_|1-153	8tlnE|1-155	4tmnE|1-155	1npc_|1-156	1ddt_|1-187
1ast_|W.C.	1iag_|W.C.	1at1A|W.C.	1kapP|1-239	1sat_|4-239	2msbA|W.C.
1hfc_|W.C.	1mnc_|W.C.	1mmq_|W.C.	2srt_|W.C.	1fbl_|100-271	1smnA|W.C.
1lkkA|W.C.	1shaA|W.C.	1shdA|W.C.	1ayaA|W.C.	1griA|64-156	1ordA|570-730
2pna_|W.C.	1ab2_|W.C.	2pldA|W.C.	1hid_|W.C.	1ptf_|W.C.	1yua_|1-65
1poh_|W.C.	1pch_|W.C.	1zer_|W.C.	1gtqA|W.C.	1puc_|W.C.	1vcc_|W.C.
1cksA|W.C.	1dksA|W.C.	1sryA|111-421	1lylA|161-502	1asyA|205-557	1chmA|157-402
1bia_|64-270	1vil_|W.C.	1svq_|W.C.	2prf_|W.C.	1acf_|W.C.	1cyo_|W.C.
1pne_|W.C.	1pfl_|W.C.	2phy_|W.C.	1mut_|W.C.	1tlcA|W.C.	1lba_|W.C.
1tsv_|W.C.	4tms_|W.C.	1tis_|W.C.			

31 multi (μ) domains

1cdkA|W.C.	1daaA|W.C.	1mml_|W.C.	1spiA|W.C.	1bucA|1-232	4blmA|W.C.
1hleA|W.C.	2cpkE|W.C.	1ckiA|W.C.	1vrtA|4-429	2hhmA|W.C.	3mddA|11-241
1athA|W.C.	1ovaA|W.C.	2achA|W.C.	1csn_|W.C.	1lgr_|W.C.	1inp_|W.C.
2bltA|W.C.	3pte_|W.C.	1btl_|W.C.	9apiA|W.C.	1irk_|W.C.	1ecl_|W.C.
5fbpA|W.C.	8catA|W.C.	1cae_|W.C.	3blm_|W.C.	1attA|W.C.	1ftaA|W.C.
1kfd_|519-928					

168 small protein (σ) domains

6rlxA|W.C.	1cphA|W.C.	1trzA|W.C.	3insA|W.C.	2gf1_|W.C.	2drpA|103-139
1bomA|W.C.	1etl_|W.C.	1wgtA|1-52	1hev_|W.C.	1mmc_|W.C.	1pyiA|30-71
1mctI|W.C.	1ppeI|W.C.	4cpaI|W.C.	2eti_|W.C.	1kal_|W.C.	1hra_|W.C.
1omc_|W.C.	1omn_|W.C.	1omg_|W.C.	1oma_|W.C.	1eit_|W.C.	1rdg_|W.C.
2sn3_|W.C.	1vna_|W.C.	1nra_|W.C.	1ptx_|W.C.	1mtx_|W.C.	1ragB|101-153
1sxm_|W.C.	2crd_|W.C.	1scy_|W.C.	1agt_|W.C.	1chl_|W.C.	1dmc_|W.C.
1sis_|W.C.	1pnh_|W.C.	1ktx_|W.C.	1ica_|W.C.	1gpt_|W.C.	1ard_|W.C.
1gps_|W.C.	11pbA|6-44	1bi6H|8-31	1tabI|W.C.	1pmc_|W.C.	1c1d_|W.C.
3ebx_|W.C.	1tgxA|W.C.	1fas_|W.C.	1ntn_|W.C.	1cdtA|W.C.	1aaf_|W.C.
2ctx_|W.C.	1lsi_|W.C.	1tfs_|W.C.	1abtA|W.C.	1kbaA|W.C.	6rxn_|W.C.
2cdx_|W.C.	2ccx_|W.C.	1cre_|W.C.	2crs_|W.C.	1cod_|W.C.	1chc_|W.C.
1nea_|W.C.	1ntx_|W.C.	1nor_|W.C.	1drs_|W.C.	1erg_|W.C.	1adn_|W.C.
1bpi_|W.C.	4tpiI|W.C.	1bpt_|W.C.	1aapA|W.C.	1knt_|W.C.	1znf_|W.C.

1dtx_|W.C.	1bunB|W.C.	1shp_|W.C.	1dtk_|W.C.	1dem_|W.C.	1gatA|W.C.
1tap_|W.C.	1dfnA|W.C.	1bnb_|W.C.	1bds_|W.C.	1sh1_|W.C.	1mea_|W.C.
1atx_|W.C.	1ah1_|W.C.	1ans_|W.C.	1ldl_|W.C.	1esl_|119-156	1iro_|W.C.
1hcgB|W.C.	1apo_|W.C.	1pth_|33-73	1egf_|W.C.	2tgf_|W.C.	1mhu_|W.C.
1ixa_|W.C.	1urk_|6-49	1tpg_|51-91	1hre_|W.C.	1zaq_|W.C.	1ptq_|W.C.
1cnr_|W.C.	1bhp_|W.C.	2plh_|W.C.	1pk4_|W.C.	1tpkA|W.C.	1bbo_|1-28
1ceaA|W.C.	2pf2_|1-65	2hppP|W.C.	1kdu_|W.C.	1fbr_|1-46	1latA|W.C.
1tpg_|W.C.	1sgpI|W.C.	3ovo_|W.C.	1hpt_|W.C.	1tgsI|W.C.	1tfi_|W.C.
1bus_|W.C.	1pce_|W.C.	4sgbI|W.C.	1tih_|W.C.	1pspA|1-53	1caa_|W.C.
1pdgA|W.C.	2tgi_|W.C.	1bndA|W.C.	1bet_|W.C.	1hcnA|W.C.	1mrb_|W.C.
1hfh_|1-63	1tcg_|W.C.	2ech_|W.C.	1fvl_|W.C.	1kst_|W.C.	1d66A|8-48
1edn_|W.C.	1srb_|W.C.	1ahtI|W.C.	1ihsI|W.C.	1fphI|W.C.	1hcqA|W.C.
2hgtI|W.C.	1dec_|W.C.	2bbkL|W.C.	2madL|W.C.	1pdc_|W.C.	8rxnA|W.C.
1ata_|W.C.	1ncfA|11-70	1afp_|W.C.	2cy3_|W.C.	2cdv_|W.C.	4at1B|101-153
1isuA|W.C.	1hip_|W.C.	2hipA|W.C.	1hpi_|W.C.	1zaaC|W.C.	4mt2_|W.C.
39 peptides (ρ) domains					
1grmA|W.C.	3aahB|W.C.	1sut_|W.C.	1smfI|W.C.	1gna_|W.C.	1lyp_|W.C.
1aml_|W.C.	1bba_|W.C.	193dC|W.C.	1cfh_|W.C.	1aty_|W.C.	1paj_|W.C.
1psm_|W.C.	1rpv_|W.C.	1ppt_|W.C.	185dA|W.C.	1ale_|W.C.	1 bdk_|W.C.
1pan_|W.C.	2mltA|W.C.	1cfg_|W.C.	2dtb_|W.C.	1sol_|W.C.	1fct_|W.C.
1bha_|W.C.	1kb7_|W.C.	1ter_|W.C.	1hph_|W.C.	2da8A|W.C.	1alf_|W.C.
1vtp_|W.C.	1btq_|W.C.	1rpc_|W.C.	1wfbA|W.C.	1gcn_|W.C.	1tor_|W.C.
1tvs_|W.C.	1plp_|W.C.	1spf_|W.C.			

Table A.5 The 225 Protein Domains.

61 all-α domains					
3sdhA|W.C.	1grj_|2-79	1ycc_|W.C.	1enh_|W.C.	1dtr_|65-191	2tct_|W.C.
1tns_|W.C.	2spcA|W.C.	1fc2C|W.C.	1bal_|W.C.	1erl_|W.C.	1rtm1|73-104
1acp_|W.C.	2abd_|W.C.	2end_|W.C.	1lis_|W.C.	1aab_|W.C.	1fps_|W.C.
1mmoG|W.C.	1lpe_|W.C.	1bcfA|W.C.	1rhgA|W.C.	1ropA|W.C.	1oelA|2-136
1eciA|W.C.	1octC|5-75	1coo_|W.C.	1mdyA|W.C.	4icb_|W.C.	1ecmA|W.C.
1parA|W.C.	1dsbA|65-128	2gstA|85-217	1bmtA|651-740	1c5a_|W.C.	2sblB|150-838
1hyp_|W.C.	2bpfA|9-91	1olgA|W.C.	1hnr_|W.C.	1aep_|W.C.	1poc_|W.C.
1axn_|W.C.	1tadA|57-177	1ezm_|154-298	2ts1_|228-319	2hmx_|W.C.	1bvp1|1-120
1llp_|W.C.	2abk_|W.C.	1gln_|306-468	1lla_|2-379	2pgd_|177-473	1aorA|211-605
1utg_|W.C.	1glm_|W.C.	1csh_|W.C.	1phc_|W.C.	1fiaA|W.C.	2wrpR|W.C.
1prcC|W.C.					
45 all-β domains					
1ddt_|381-535	1cdg_|582-686	1hoe_|W.C.	1aac_|W.C.	2bpa1|W.C.	2pec_|W.C.
4gcr_|1-85	2sblB|7-149	1pgs_|4-140	1gof_|1-150	1bvp1|1-217	1lxa_|W.C.
1knb_|W.C.	1tnrA|W.C.	1thw_|W.C.	1scs_|W.C.	1bglA|731-1023	2phlA|W.C.
1bia_|271-317	1ltsD|W.C.	1prcH|W.C.	4fgf_|W.C.	1fnb_|19-154	1htp_|W.C.
1eft_|313-405	1arb_|W.C.	1bco_|481-560	1difA|W.C.	1gtrA|339-547	2kauC|2-129
1btn_|W.C.	1pkn_|116-217	1hbq_|W.C.	1sriA|W.C.	2cpl_|W.C.	1kapP|247-470
1hxn_|W.C.	2sil_|W.C.	1gof_|151-537	3aahA|W.C.	1cdg_|383-495	1msaA|W.C.
1hcb_|W.C.	3bcl_|W.C.	1vmoA|W.C.			
56 α/β domains					
1cdg_|1-382	1tml_|W.C.	2tmdA|490-645	1dik_|377-505	1bta_|W.C.	3ecaA|W.C.
1bnh_|W.C.	1iceA|W.C.	1mla_|3-127	1fnb_|155-314	2ts1_|1-217	1ctt_|1-150
2tmdA|341-489	2ohxA|175-324	1bncA|1-114	1pydA|2-181	1nbaA|W.C.	1pfkA|W.C.
1deaA|W.C.	1gky_|W.C.	1chd_|W.C.	1cseE|W.C.	3cla_|W.C.	2dri_|W.C.
1phr_|W.C.	2hnq_|W.C.	2trxA|W.C.	1trkA|535-680	1pkm_|396-530	2olbA|W.C.
1lcpA|1-159	1eriA|W.C.	2rslA|W.C.	1hpm_4-188	3pgm_|W.C.	2bgu_|W.C.
1gph1235-465	1lfaA|W.C.	1garA|W.C.	1hmy_|W.C.	7aatA|W.C.	1pxtA|28-293
1ulb_|W.C.	1gpmA|3-207	1ack_|W.C.	2ctb_|W.C.	8dfr_|W.C.	1mioA|W.C.
1aliA|W.C.	1xaa_|W.C.	4at1A|1-150	1ubsB|W.C.	1ora_|1-149	3pgk_|W.C.
7acn_|2-528	3pmgA|1-190				
63 α+β domains					
1fus_|W.C.	1cnsA|W.C.	2act_|W.C.	7rsa_|W.C.	2kauA|W.C.	1mrj_|W.C.
1napA|W.C.	1sso_|W.C.	1pkp_|78-147	1igd_|W.C.	3cox_|319-450	1ltsA|W.C.
1molA|W.C.	1fruA|1-178	1aak_|W.C.	1fkd_|W.C.	1dhy_|1-132	1esl_|1-118
1cseI|W.C.	1mngA|93-203	1ctf_|W.C.	2reb_|269-328	1stu_|W.C.	1mldA|145-313
1vih_|W.C.	1gpmA|405-525	2mnr_|3-132	1oelA|137-190	1fxd_|W.C.	1pyaA|W.C.
1tig_|W.C.	1kptA|W.C.	3rubS|W.C.	1dchA|W.C.	2chsA|W.C.	1gph1|1-234
1otfA|W.C.	1gadO|149-312	1oacA|5-90	3sicI|W.C.	2ms2A|W.C.	1aorA|1-210
3grs_|364-478	1ezm_|1-153	1lkkA|W.C.	1hid_|W.C.	1puc_|W.C.	2dnjA|W.C.
1sryA|111-421	1vil_|W.C.	2prf_|W.C.	1mut_|W.C.	1tlcA|W.C.	2glt_|123-316
1lba_|W.C.	1cyo_|W.C.	1vcc_|W.C.	1ordA|570-730	1smnA|W.C.	2polA|1-122
1chmA|157-402	1ytbA|61-155	1xrb_|1-101			

Table A.6 The 510 Protein Domains.

109 all-α domains

1sctA|W.C.	1ytc_|W.C.	1yea_|W.C.	1yeb_|W.C.	2pccB|W.C.	1phd_|W.C.
1fhb_|W.C.	1cih_|W.C.	1cie_|W.C.	1csu_|W.C.	1crj_|W.C.	1noo_|W.C.
1csw_|W.C.	1csx_|W.C.	1cri_|W.C.	1chi_|W.C.	1cig_|W.C.	1grl_|6-136
1crh_|W.C.	1raq_|W.C.	1ctz_|W.C.	1chj_|W.C.	1cif_|W.C.	1phg_|W.C.
1csv_|W.C.	1crg_|W.C.	1chh_|W.C.	1cty_|W.C.	1rap_|W.C.	3fisA|W.C.
1hddC|W.C.	1dprA|65-136	1tnt_|W.C.	1bbl_|W.C.	1erc_|W.C.	1afb1|73-104
1aca_|W.C.	1vasA|W.C.	1enj_|W.C.	1enk_|W.C.	1eni_|W.C.	1phf_|W.C.
1lynA|W.C.	1hme_|W.C.	1hmf_|W.C.	1hsm_|W.C.	1hsn_|W.C.	1fipA|W.C.
1nhm_|W.C.	1nhn_|W.C.	1gnc_|W.C.	1rprA|W.C.	1rpo_|W.C.	1afa1|73-104
1pou_|W.C.	1cdn_|W.C.	1bod_|W.C.	1boc_|W.C.	2bca_|W.C.	1phe_|W.C.
2bcb_|W.C.	1clb_|W.C.	1arqA|W.C.	1arrA|W.C.	1mykA|W.C.	1troA|W.C.
1mylA|W.C.	1bpd_|9-91	2bpgA|9-91	1olhA|W.C.	1pesA|W.C.	1afd1|73-104
1petA|W.C.	1seaA|W.C.	1safA|W.C.	1sagA|W.C.	1sahA|W.C.	1cp4_|W.C.
1saiA|W.C.	1sajA|W.C.	1sakA|W.C.	1salA|W.C.	1hns_|W.C.	1trrA|W.C.
1tag_|57-177	1tndA|57-177	1tyc_|228-319	1tydE|228-319	1tybE|228-319	1pha_|W.C.
1tyaE|228-319	1lgaA|W.C.	1oxy_|1-379	1nol_|1-279	1pgn_|177-473	2cpp_|W.C.
1pgo_|177-473	1pgp_|177-473	1pgg_|177-473	3gly_|W.C.	1dog_|W.C.	1phb_|W.C.
1agm_|W.C.	1csi_|W.C.	1css_|W.C.	1csr_|W.C.	1csc_|W.C.	5cscA|W.C.
5cts_|W.C.					

130 all-β domains

1mdtA|381-535	1cgt_|580-684	1cxe_|582-686	1cxi_|582-686	1cxg_|582-686	1cxi_|383-495
1cxh_|582-686	1cxf_|582-686	1cgv_|582-686	1cgw_|582-686	1cgy_|582-686	1cgw_|383-495
1cgx_|582-686	1cgu_|580-684	1aaj_|W.C.	1aan_|W.C.	2mtaA|W.C.	1crm_|W.C.
1mdaA|W.C.	1gcs_|1-85	1pnf_|1-140	1png_|5-140	1gog_|1-150	1akl_|247-470
1goh_|1-150	1tnfA|W.C.	2tunA|W.C.	1thv_|W.C.	1thu_|W.C.	1cxg_|383-495
2ctvA|W.C.	1scr_|W.C.	1conA|W.C.	5cnaA|W.C.	1apnA|W.C.	1cgy_|383-495
2cna_|W.C.	1cn1A|W.C.	1bib_|271-317	1ltaD|W.C.	1lttD|W.C.	1azm_|W.C.
1ltgD|W.C.	1ltbD|W.C.	1htlD|W.C.	1bfb_|W.C.	1bfc_|W.C.	1hpcA|W.C.
1fga_|W.C.	2bfh_|W.C.	1bfg_|W.C.	1bas_|W.C.	1fnd_|19-154	1cxh_|383-495
1fnc_|19-154	1frn_|19-154	1arc_|W.C.	1bcmA|481-560	1hpxA|W.C.	1cgx_|383-495
1hihA|W.C.	1hvjA|W.C.	1hvkA|W.C.	1hivA|W.C.	1hpvA|W.C.	1bzm_|W.C.
1hsgA|W.C.	1hshA|W.C.	1hvlA|W.C.	1cpiA|W.C.	1hvrA|W.C.	1kraC|2-129
1htgA|W.C.	1hvc_|W.C.	4phvA|W.C.	1hosA|W.C.	1sbgA|W.C.	1cxf_|383-495
1hhp_|W.C.	5hvpA|W.C.	1hbvA|W.C.	1hefE|W.C.	1hpsA|W.C.	1cgu_|383-494
1hsiA|W.C.	1hegE|W.C.	1aaqA|W.C.	1htfA|W.C.	1hteA|W.C.	1czm_|W.C.
3hvp_|W.C.	3phv_|W.C.	1hvsA|W.C.	1gtsA|339-547	1hbp_|W.C.	1krbC|2-129
1fen_|W.C.	1erb_|W.C.	1fel_|W.C.	1fem_|W.C.	1slfB|W.C.	1cgv_|383-495
1srgA|W.C.	1sreA|W.C.	1srjA|W.C.	1slgB|W.C.	1ptsA|W.C.	1hug_|W.C.
1sleB|W.C.	1srfA|W.C.	1strB|W.C.	1stsB|W.C.	1sldB|W.C.	1huh_|W.C.
1srhA|W.C.	1stp_|W.C.	1cyhA|W.C.	1mikA|W.C.	2rmaA|W.C.	1krcC|2-129
1cwaA|W.C.	1cwcA|W.C.	2rmbA|W.C.	1cwbA|W.C.	3cysA|W.C.	1cxe_|383-495
2sim_|W.C.	1gog_|151-537	1goh_|151-537	1cgt_|383-494		

135 α/β domains

1cgt_|1-382	1cxe_|1-382	1cxi_|1-382	1cxg_|1-382	1cxh_|1-382	1racA|1-150
1cxf_|1-382	1cgv_|1-382	1cgw_|1-382	1cgy_|1-382	1cgx_|1-382	1rahA|1-150
1cgu_|1-382	1btb_|W.C.	1brsD|W.C.	1bgsE|W.C.	1fnd_|155-314	1wsyB|W.C.
1fnc_|155-314	1frn_|155-314	1tyc_|1-217	1tydE|1-217	1tybE|1-217	1dbp_|W.C.

1tyaE|1-217	1cdoA|176-324	1hldA|175-324	2oxiA|175-324	1adbA|175-324	1radA|1-150
1adg_|175-324	1adf_|175-324	8adh_|175-324	1adcA|175-324	6adhA|175-324	8atcA|1-150
1horA|W.C.	1hotA|W.C.	2secE|W.C.	1sca_|W.C.	1scnE|W.C.	1orb_|1-149
1scd_|W.C.	1scb_|W.C.	1sbc_|W.C.	1selA|W.C.	1cia_|W.C.	1drj_|W.C.
1pnt_|W.C.	1bvh_|W.C.	2hnp_|W.C.	2tir_|W.C.	1tho_|W.C.	1raeA|1-150
1tkbA|535-680	1tkcA|535-680	1tkaA|535-680	1lam_|1-159	1lanA|1-159	1acmA|1-150
1bllE|1-159	1lap|1-159	1bpm_|1-159	1bpn_|1-159	1gdtA|1-140	1rhd_|1-149
3hsc_|3-188	1ngj_|3-188	1ngi_|4-188	1ngb_|4-188	1ngf_|3-188	1olcA|W.C.
1nga_|4-188	1nge_|4-188	1ngc_|4-188	1ngg_|3-188	1ngh_|4-188	1rafA|1-150
1atr_|2-188	1ngd_|4-188	1ats_|2-188	1cde_|W.C.	1grcA|W.C.	1ttqB|W.C.
1cddA|W.C.	1mhtA|W.C.	1ama_|W.C.	1maq_|W.C.	1tarA|W.C.	2bgt_|W.C.
1map_|W.C.	1tasA|W.C.	1tatA|W.C.	1akaA|W.C.	1akbA|W.C.	1olaA|W.C.
1akcA|W.C.	1ula_|W.C.	1amn_|W.C.	1acj_|W.C.	1acl_|W.C.	1ragA|1-150
1ace_|W.C.	2ctc_|W.C.	5cpa_|W.C.	1cbx_|W.C.	1cps_|W.C.	1ttpB|W.C.
1dr1_|W.C.	1dr3_|W.C.	1dr2_|W.C.	1dr6_|W.C.	1dr4_|W.C.	1drk_|W.C.
1dr5_|W.C.	1dr7_|W.C.	2anhA|W.C.	1hqaA|W.C.	1alkA|W.C.	1ctu_|1-150
1ajaA|W.C.	1ajdA|W.C.	1anjA|W.C.	1a1jA|W.C.	1aniA|W.C.	1rabA|1-150
1alhA|W.C.	1ajbA|W.C.	1ajcA|W.C.	1xab_|W.C.	1ipd_|W.C.	1raaA|1-150
1hex_|W.C.	1idm_|W.C.	1raiA|1-150			

136 α+β domains

1fut_|W.C.	1rck_|W.C.	1rcl_|W.C.	2baa_|W.C.	1aec_|W.C.	3tms_|W.C.
2rat_|W.C.	1rpg_|W.C.	1rhb_|W.C.	1rnc_|W.C.	2rns_|W.C.	3b5c_|W.C.
1rnd_|W.C.	3rn3_|W.C.	1rbx_|W.C.	1rob_|W.C.	1rnu_|W.C.	3dni_|W.C.
1ras_|W.C.	1rnv_|W.C.	1rnnE|W.C.	1rno_|W.C.	1rar_|W.C.	1tcs_|W.C.
1rbw_|W.C.	1rnmE|W.C.	1rha_|W.C.	1rsm_|W.C.	1rbn_|W.C.	1htlA|W.C.
1rnq_|W.C.	1sscA|W.C.	1ssbA|W.C.	1rca_|W.C.	1srnA|W.C.	1tsx_|W.C.
1rpf_|W.C.	1rph_|W.C.	1rbbA|W.C.	1rcnE|W.C.	1rtaE|W.C.	1tbpA|61-155
1rtb_|W.C.	1rbjA|W.C.	1rbh_|W.C.	2aas_|W.C.	1rbd_|W.C.	1dnkA|W.C.
1rbi_|W.C.	2rlnE|W.C.	1kraA|W.C.	1rbe_|W.C.	1rbg_|W.C.	1ltaA|W.C.
1rbf_|W.C.	1rbc_|W.C.	1pga_|W.C.	1krbA|W.C.	1krcA|W.C.	1xrc_|1-101
1pgx_|W.C.	1pgb_|W.C.	2igh_|W.C.	1igcA|W.C.	1fccC|W.C.	1atnD|W.C.
1gb1_|W.C.	2igg_|W.C.	1fkb_|W.C.	1coy_|319-450	3monA|W.C.	1lttA|W.C.
1frtA|1-178	1fkj_|W.C.	1fkg_|W.C.	1fkf_|W.C.	1fkl_|W.C.	1tsv_|W.C.
2fke_|W.C.	1fkh_|W.C.	1fkt_|W.C.	1fkk_|W.C.	1fkiA|W.C.	1xra_|1-101
1fkr_|W.C.	1fks_|W.C.	1acbI|W.C.	2secI|W.C.	1egpA|W.C.	4mdhA|155-333
1meeI|W.C.	2tecI|W.C.	1vig_|W.C.	1egl_|W.C.	1sbnI|W.C.	1ltgA|W.C.
1sibI|W.C.	3mdsA|93-203	1r1cS|W.C.	1mns_|3-132	1mdr_|3-132	1tys_|W.C.
1grl_|137-190	1rldS|W.C.	1bmsA|W.C.	1comA|W.C.	2chtA|W.C.	1glv_|123-316
1gaeO|149-312	1mstA|W.C.	1grf_|364-478	1msc_|W.C.	1grb_|364-478	1mrk_|W.C.
1gra_|364-478	1gre_|364-478	1lckA|117-226	1grg_|364-478	4grl_|364-478	1ltbA|W.C.
1lklA|W.C.	1lcjA|W.C.	1sesA|111-421	1sphA|W.C.	2hpr_|W.C.	1tsw_|W.C.
1sceA|W.C.	1setA|111-421	1synA|W.C.	1serA|111-421	2tscA|W.C.	1tsy_|W.C.
1tsdA|W.C.	2bbqA|W.C.	1tsz_|W.C.	1ssaA|W.C.		

Table A.7 The 2438 Protein Domains.

393 all-α domains					
1aca_|W.C.	1cpe_|W.C.	1hbbB|W.C.	1mbi_|W.C.	1pra_|W.C.	2dhbA|W.C.
1afb1|73-104	1cpf_|W.C.	1hbhA|W.C.	1mbj_|W.C.	1prhA|74-586	2dhbB|W.C.
1afd1|73-104	1cpg_|W.C.	1hbhB|W.C.	1mbk_|W.C.	1pru_|W.C.	2fal_|W.C.
1agm_|W.C.	1crcA|W.C.	1hbsA|W.C.	1mbo_|W.C.	1prv_|W.C.	2fam_|W.C.
1anwA|W.C.	1crg_|W.C.	1hbsB|W.C.	1mcy_|W.C.	1pvaA|W.C.	2frc_|W.C.
1anxA|W.C.	1crh_|W.C.	1hc1_|W.C.	1mgn_|W.C.	1r36_|W.C.	2glrA|79-209
1apc_|W.C.	1cri_|W.C.	1hc3_|5-398	1mlf_|W.C.	1rap_|W.C.	2hbdA|W.C.
1arp_|W.C.	1crj_|W.C.	1hc4_|W.C.	1mlg_|W.C.	1raq_|W.C.	2hbdB|W.C.
1arqA|W.C.	1csgA|W.C.	1hc5_|W.C.	1mlh_|W.C.	1rcc_|W.C.	2hbeA|W.C.
1arrA|W.C.	1csi_|W.C.	1hc6_|5-398	1mlj_|W.C.	1rce_|W.C.	2hbeB|W.C.
1arv_|W.C.	1csr_|W.C.	1hcy_|W.C.	1mlk_|W.C.	1rcg_|W.C.	2hbfA|W.C.
1arw_|W.C.	1css_|W.C.	1hdbA|W.C.	1mll_|W.C.	1rci_|W.C.	2hbfB|W.C.
1arx_|W.C.	1csu_|W.C.	1hdbB|W.C.	1mlm_|W.C.	1res_|W.C.	2hcoA|W.C.
1ary_|W.C.	1csx_|W.C.	1hgaA|W.C.	1mln_|W.C.	1ret_|W.C.	2hcoB|W.C.
1avhA|W.C.	1ctaA|W.C.	1hgaB|W.C.	1mlo_|W.C.	1rnrA|W.C.	2hhdA|W.C.
1avr_|W.C.	1ctdA|W.C.	1hgbA|W.C.	1mlq_|W.C.	1rpo_|W.C.	2hhdB|W.C.
1aypA|W.C.	1ctr_|W.C.	1hgbB|W.C.	1mlr_|W.C.	1rprA|W.C.	2hmqA|W.C.
1babA|W.C.	1cty_|W.C.	1hgcA|W.C.	1mlu_|W.C.	1saeA|W.C.	2hoa_|W.C.
1babB|W.C.	1ctz_|W.C.	1hgcB|W.C.	1mnh_|W.C.	1safA|W.C.	2ifn_|W.C.
1bbbA|W.C.	1cxa_|W.C.	1hgu_|W.C.	1mnjA|W.C.	1sagA|W.C.	2int_|W.C.
1bbbB|W.C.	1cyf_|W.C.	1hhoA|W.C.	1mnkA|W.C.	1sahA|W.C.	2lh2_|W.C.
1bbl_|W.C.	1cyj_|W.C.	1hhoB|W.C.	1moa_|W.C.	1saiA|W.C.	2lh6_|W.C.
1bbn_|W.C.	1cyl_|W.C.	1hij_|W.C.	1mob_|W.C.	1sajA|W.C.	2lh7_|W.C.
1bcn_|W.C.	1dcc_|W.C.	1hik_|W.C.	1moc_|W.C.	1salA|W.C.	2mgb_|W.C.
1bgd_|W.C.	1dog_|W.C.	1hkt_|W.C.	1mrrA|W.C.	1san_|W.C.	2mgc_|W.C.
1bib_|2-63	1dprA|3-64	1hlm_|W.C.	1msdA|1-83	1sctA|W.C.	2mgd_|W.C.
1boc_|W.C.	1dprA|65-136	1hmdA|W.C.	1mti_|W.C.	1sesA|W.C.	2mge_|W.C.
1bod_|W.C.	1dvh_|W.C.	1hme_|W.C.	1mtj_|W.C.	1setA|W.C.	2mgf_|W.C.
1bpd_|W.C.	1dxtA_|W.C.	1hmf_|W.C.	1mtk_|W.C.	1spe_|W.C.	2mgg_|W.C.
1bpq_|W.C.	1dxtB|W.C.	1hmoA|W.C.	1myf_|W.C.	1swm_|W.C.	2mgh_|W.C.
1bvd_|W.C.	1dxuA|W.C.	1hnbA|85-217	1myhA|W.C.	1tag_|W.C.	2mgi_|W.C.
1cblA|W.C.	1dxuB|W.C.	1hncA|W.C.	1myiA|W.C.	1thbA|W.C.	2mgj_|W.C.
1cblB|W.C.	1dxvA|W.C.	1hns_|W.C.	1myjA|W.C.	1thbB|W.C.	2mgk_|W.C.
1cbmA|W.C.	1dxvB|W.C.	1hrm_|W.C.	1mykA|W.C.	1thl_|W.C.	2mgl_|W.C.
1cbmB|W.C.	1ecd_|W.C.	1hsm_|W.C.	1mylA|W.C.	1tlpE|156-316	2mgm_|W.C.
1cca_|W.C.	1ecn_|W.C.	1hsn_|W.C.	1mym_|W.C.	1tndA|W.C.	2mm1_|W.C.
1ccb_|W.C.	1eco_|W.C.	1hsy_|W.C.	1ner_|W.C.	1tnp_|W.C.	2mya_|W.C.
1cce_|W.C.	1eni_|W.C.	1huw_|W.C.	1nhm_|W.C.	1tnq_|W.C.	2myb_|W.C.
1ccg_|W.C.	1enj_|W.C.	1hve_|W.C.	1nhn_|W.C.	1tnt_|W.C.	2myd_|W.C.
1cch_|W.C.	1enk_|W.C.	1hvf_|W.C.	1nihA|W.C.	1tnw_|W.C.	2mye_|W.C.
1ccp_|W.C.	1erc_|W.C.	1hvg_|W.C.	1nihB|W.C.	1tnx_|W.C.	2pac_|W.C.
1cdlA_|W.C.	1esp_|W.C.	1hyt_|156-316	1nol_|W.C.	1trf_|W.C.	2pas_|W.C.
1cdmA_|W.C.	1fcs_|W.C.	1ifd_|W.C.	1noo_|W.C.	1trlA|W.C.	2pcbA|W.C.
1cdn_|W.C.	1fhb_|W.C.	1ifi_|W.C.	1olhA|W.C.	1tyaE|228-319	2pcbB|W.C.
1ceh_|W.C.	1fipA|W.C.	1isbA|1-82	1omd_|W.C.	1tybE|228-319	2pccA|W.C.
1cfc_|W.C.	1fw4_|W.C.	1iscA|1-82	1oxy_|1-379	1tyc_|W.C.	2pccB|W.C.

1chh_|W.C.	1gclA|W.C.	1iti_|W.C.	1pesA|W.C.	1tydE_|W.C.	2pde_|W.C.
1chi_|W.C.	1gcmA|W.C.	1itm_|W.C.	1petA|W.C.	1was_|W.C.	2phiA|W.C.
1chj_|W.C.	1gdd_|W.C.	1leb_|W.C.	1pgn_|W.C.	1watA|W.C.	2spl_|W.C.
1cie_|W.C.	1gdi_|W.C.	1lgaA|W.C.	1pgo_|177-473	1ycaA|W.C.	2spm_|W.C.
1cif_|W.C.	1gdj_|W.C.	1lh3_|W.C.	1pgp_|177-473	1ycbA|W.C.	2spn_|W.C.
1cig_|W.C.	1gdk_|W.C.	1lh5_|W.C.	1pgq_|W.C.	1yea_|W.C.	2spo_|W.C.
1cih_ |W.C.	1gdl_|W.C.	1lhs_|W.C.	1pha_|W.C.	1yeb_|W.C.	2sttA|W.C.
1cib_|W.C.	1gfi_|61-181	1lih_|W.C.	1phb_|W.C.	1yma_|W.C.	3fisA|W.C.
1clm_|W.C.	1gil_|61-181	1lnaE|156-316	1phd_|W.C.	1ymc_|W.C.	3gly_|W.C.
1cmcA|W.C.	1glpA|79-209	1lnbE_|W.C.	1phe_|W.C.	1ytc_|W.C.	3hsf_|W.C.
1cmf_|W.C.	1gnc_|W.C.	1lncE_|W.C.	1phf_|W.C.	2bbmA|W.C.	3inkC|W.C.
1cmg_|W.C.	1gne_|80-232	1lndE_|W.C.	1phg_|W.C.	2bbnA|W.C.	2mdeA|242-395
1cmp_|W.C.	1grl_|6-136	1lneE_|156-316	1pir_|W.C.	2bca_|W.C.	3mdsA|W.C.
1cmq_|W.C.	1gsdA|W.C.	1lnfE_|W.C.	1pis_|W.C.	2bcb_|W.C.	3pat_|W.C.
1cmt_|W.C.	1gsfA|81-222	1lynA_|W.C.	1pmbA|W.C.	2bmhA|W.C.	4cpv_|W.C.
1cmu_|W.C.	1gtb_|W.C.	1mbc_|W.C.	1pobA|W.C.	2bpp_|W.C.	4mbn_|W.C.
1cmyA|W.C.	1guhA|W.C.	1mbd_|W.C.	1pod_|W.C.	2cep_|W.C.	155c_|W.C.
1cmyB|W.C.	1hbaA|W.C.	1mbf_|W.C.	1poeA|W.C.	2cmm_|W.C.	
1copD|W.C.	1hbaB|W.C.	1mbg_|W.C.	1pog_|W.C.	2cxbA|W.C.	
1cp4_|W.C.	1hbbA|W.C.	1mbh_|W.C.	1pou_|W.C.	2cyk_|W.C.	

704 all-β domains

1aaj_|W.C.	1cpiA|W.C.	1gmcA|W.C.	1krcB|W.C.	1plb_|W.C.	1vfbA|W.C.
1aan_|W.C.	1cpm_|W.C.	1gmdA|W.C.	1krcC|2-129	1pnc_|W.C.	1xnc_|W.C.
1aaqA|W.C.	1cra_|W.C.	1gmh_|W.C.	1krt_|W.C.	1pnd_|W.C.	1xypA|W.C.
1abiL|W.C.	1crb_|W.C.	1gog_|1-150	1lac_|W.C.	1pnf_|1-140	1yda_|W.C.
1abq_|W.C.	1crm_|W.C.	1gog_|151-537	1lec_|W.C.	1png_|5-140	1ydb_|W.C.
1acbE|W.C.	1csq_|W.C.	1gog_|538-639	1lemA|W.C.	1pnj_|W.C.	1ydc_|W.C.
1adbA|1-174	1cvb_|W.C.	1goh_|1-150	1lenA|W.C.	1ppbL|W.C.	1ydd_|W.C.
1adcA|1-174	1cvc_|W.C.	1goh_|151-537	1lgbA|W.C.	1ppgE|W.C.	1yhaA|W.C.
1adf_|1-174	1cvd_|W.C.	1goh_|538-639	1lic_|W.C.	1pphE|W.C.	1yhb_|W.C.
1adg_|1-174	1cve_|W.C.	1hagE|W.C.	1lid_|W.C.	1ppkE|W.C.	1ystH|36-260
1adl_|W.C.	1cvf_|W.C.	1haiL|W.C.	1lie_|W.C.	1pplE|W.C.	2azaA|W.C.
1afcA|W.C.	1cvh_|W.C.	1hapL|W.C.	1lif_|W.C.	1prlC|W.C.	2bat_|W.C.
1aizA|W.C.	1cwaA|W.C.	1hbp_|W.C.	1loaA|W.C.	1prmC|W.C.	2bfh_|W.C.
1akl_|247-470	1cwbA|W.C.	1hbtL|W.C.	1lobA|W.C.	1prs1|W.C.	2cab_|W.C.
1alb_|W.C.	1cwcA|W.C.	1hbvA|W.C.	1locA|W.C.	1psaA|W.C.	2cbb_|W.C.
1apnA|W.C.	1cxe_|383-495	1hc1_|399-653	1lodA|W.C.	1pse_|W.C.	2cbc_|W.C.
1aptE|W.C.	1cxe_|496-581	1hc3_|399-653	1lofA|W.C.	1psn_|W.C.	2cbd_|W.C.
1apuE|W.C.	1cxe_|582-686	1hc4_|399-653	1logA|W.C.	1pssH|36-248	2cbe_|W.C.
1apvE|W.C.	1cxf_|582-686	1hc5_|399-653	1lpaB|337-449	1pstH|36-248	2cgaA|W.C.
1apwE|W.C.	1cxf_|383-495	1hc6_|399-653	1ltaD|W.C.	1ptoB|88-197	2cha_|W.C.
1arc_|W.C.	1cxf_|496-581	1hca_|W.C.	1ltbD|W.C.	1ptoD|W.C.	2chbD|W.C.
1asoA|1-129	1cxg_|383-495	1hcd_|W.C.	1ltgD|W.C.	1ptoF|W.C.	2cna_|W.C.
1aspA|1-129	1cxg_|582-686	1hcy_|399-653	1lttD|W.C.	1ptsA|W.C.	2ctvA|W.C.
1asqA|1-129	1cxh_|383-495	1hdtL|W.C.	1lyaA|W.C.	1pza_|W.C.	1cxg_|496-581
1avdA|W.C.	1cxh_|496-581	1hdxA|1-174	1macA|W.C.	1pzb_|W.C.	2dblL|1-107
1aveA|W.C.	1cxh_|582-686	1hdyA|1-174	1maj_|W.C.	1pzc_|W.C.	2dblL|108-211
1azbA|W.C.	1cxi_|383-495	1hdzA|1-174	1mak_|W.C.	1qwfA|W.C.	2eipA|W.C.
1azm_|W.C.	1cxi_|496-581	1hea_|W.C.	1mcbA|1-111	1r091|W.C.	2enb_|W.C.
1aznA|W.C.	1cxi_|582-686	1hed_|W.C.	1mcbA|112-216	1ray_|W.C.	2er0E|W.C.
1azu_|W.C.	1cyhA|W.C.	1hefE|W.C.	1mccA|1-111	1raz_|W.C.	2er6E|W.C.
1bas_|W.C.	1cyw_|W.C.	1hegE|W.C.	1mccA|112-216	1rinA|W.C.	2er7E|W.C.

1bbrL|W.C.	1czm_|W.C.	1hgdA|W.C.	1mcfA|1-111	1rlpC|W.C.	2er9E|W.C.
1bbs_|W.C.	1dbbL|1-107	1hgeA|W.C.	1mcfA|112-216	1rlqC|W.C.	2gn5_|W.C.
1bcd_|W.C.	1dbbL|108-211	1hgfA|W.C.	1mchA|1-111	1rne_|W.C.	2gvaA|W.C.
1bcmA|481-560	1dbjL|1-107	1hggA|W.C.	1mchA|112-216	1ruc1|W.C.	2gvbA|W.C.
1bcx_|W.C.	1dbjL|108-211	1hghA|W.C.	1mciA|1-111	1rud1|W.C.	2hmb_|W.C.
1bfb_|W.C.	1dbkL|1-107	1hgjA|W.C.	1mciA|112-216	1rue1|W.C.	2hntL|W.C.
1bfc_|W.C.	1dbkL|108-211	1hhgA|182-275	1mcjA|1-111	1ruf1|W.C.	2hpeA|W.C.
1bfg_|W.C.	1dbmL|1-107	1hhhA|182-275	1mcjA|112-216	1rug1|W.C.	2hpfA|W.C.
1bib_|271-317	1dbmL|108-211	1hhiA|182-275	1mckA|1-111	1ruh1|W.C.	2hpqL|W.C.
1bic_|W.C.	1dca_|W.C.	1hhjA|182-275	1mckA|112-216	1rui1|W.C.	2hsp_|W.C.
1bilA|W.C.	1dcb_|W.C.	1hhkA|182-275	1mclA|1-111	1rza_|W.C.	2hwb1|W.C.
1bimA|W.C.	1dmyA|W.C.	1hhp_|W.C.	1mclA|112-216	1rzb_|W.C.	2hwc1|W.C.
1blbA|W.C.	1dwbL|W.C.	1hib_|W.C.	1mcnA|1-111	1rzc_|W.C.	2hwd1|W.C.
1bmaA|W.C.	1dwcL|W.C.	1hihA|W.C.	1mcnA|112-216	1rzd_|W.C.	2hwe1|W.C.
1bra_|W.C.	1dwdL|W.C.	1hiiA|W.C.	1mcqA|1-111	1rze_|W.C.	2hwf1|W.C.
1brcE|W.C.	1dweL|W.C.	1himL|1-108	1mcqA|112-216	1sbgA|W.C.	2ifb_|W.C.
1brp_|W.C.	1eas_|W.C.	1himL|109-211	1mcrA|1-111	1scr_|W.C.	2iffL|1-106
1brq_|W.C.	1eat_|W.C.	1hinL|1-108	1mcrA|112-216	1sdaB|W.C.	2iffL|107-212
1btb_|W.C.	1eau_|W.C.	1hinL|109-211	1mcsA|1-111	1sdyA|W.C.	2ig2L|1-109
1btwA|W.C.	1eedP|W.C.	1hivA|W.C.	1mcsA|112-216	1sgc_|W.C.	2ig2L|110-214
1btxA|W.C.	1elaA|W.C.	1hldA|1-174	1mdaA|W.C.	1sgqE|W.C.	2imn_|W.C.
1bty_|W.C.	1elbA|W.C.	1hltL|W.C.	1mdaH|W.C.	1sgrE|W.C.	2jcw_|W.C.
1btzA|W.C.	1elcA|W.C.	1hmr_|W.C.	1mdtA|381-535	1sip_|W.C.	2kaiA|W.C.
1bw4_|W.C.	1eldE|W.C.	1hmt_|W.C.	1mec1|W.C.	1slaA|W.C.	2lalA|W.C.
1byh_|W.C.	1elf_|W.C.	1hneE|W.C.	1mfcL|1-111	1slbA|W.C.	2mcg1|1-111
1bzm_|W.C.	1ena_|W.C.	1hosA|W.C.	1mfcL|112-212	1slcA|W.C.	2mcg1|112-216
1ca3_|W.C.	1enc_|W.C.	1hpcA|W.C.	1mfdL|1-111	1sldB|W.C.	2mhaA|182-270
1cah_|W.C.	1entE|W.C.	1hpsA|W.C.	1mfdL|112-212	1sleB|W.C.	2mib_|W.C.
1cai_|W.C.	1enxA|W.C.	1hpvA|W.C.	1mfeL|1-111	1slfB|W.C.	2mipA|W.C.
1caj_|W.C.	1epaA|W.C.	1hpxA|W.C.	1mfeL|112-211	1slgB|W.C.	2nrd_|8-166
1cak_|W.C.	1eplE|W.C.	1hri1|W.C.	1mikA|W.C.	1snm_|W.C.	2oxiA|1-174
1cal_|W.C.	1epmE|W.C.	1hrtL|W.C.	1mlcA|1-108	1sosA|W.C.	2pabA|W.C.
1cam_|W.C.	1epoE|W.C.	1hrv1|W.C.	1mlcA|109-214	1sreA|W.C.	2plv1|W.C.
1can_|W.C.	1eppE|W.C.	1hsgA|W.C.	1mrcL|1-108	1srfA|W.C.	2ptcE|W.C.
1cao_|W.C.	1epqE|W.C.	1hshA|W.C.	1mrcL|109-211	1srgA|W.C.	2ptn_|W.C.
1cavA|W.C.	1eprE|W.C.	1hsiA|W.C.	1mreL|1-108	1srhA|W.C.	2r041|W.C.
1cawA|W.C.	1eptA|W.C.	1htbA|1-174	1mreL|109-211	1srjA|W.C.	2r061|W.C.
1caxA|W.C.	1erb_|W.C.	1htfA|W.C.	1mrfL|1-108	1srm_|W.C.	2r071|W.C.
1cay_|W.C.	1esa_|W.C.	1htgA|W.C.	1mrfL|109-211	1srp_|247-470	2rcrH|36-255
1caz_|W.C.	1esb_|W.C.	1htlD|W.C.	1mua_|W.C.	1sta_|W.C.	2ren_|W.C.
1cbq_|W.C.	1eta1|W.C.	1hug_|W.C.	1ncbL|1-108	1stb_|W.C.	2rm21|W.C.
1cbrA|W.C.	1etb1|W.C.	1huh_|W.C.	1ncbL|109-214	1stg_|W.C.	2rmaA|W.C.
1ccs_|W.C.	1etrL|W.C.	1hva_|W.C.	1ncbN|W.C.	1sth_|W.C.	2rmbA|W.C.
1cct_|W.C.	1etsL|W.C.	1hvc_|W.C.	1nccL|1-108	1stn_|W.C.	2rmu1|W.C.
1ccu_|W.C.	1ettL|W.C.	1hviA|W.C.	1nccL|109-214	1stp_|W.C.	2rr11|W.C.
1cdb_|W.C.	1exh_|W.C.	1hvjA|W.C.	1nccN|W.C.	1strB|W.C.	2rs11|W.C.
1cdh_|1-97	1faiL|1-108	1hvkA|W.C.	1ncdL|1-108	1sxaA|W.C.	2rs31|W.C.
1cdh_|98-178	1faiL|109-214	1hvrA|W.C.	1ncdL|109-211	1sxBA|W.C.	2rs51|W.C.
1cdoA|1-175	1fccA|238-341	1hvsA|W.C.	1ncdN|W.C.	1syb_|W.C.	2sam_|W.C.
1cgiE|W.C.	1fel_|W.C.	1icm_|W.C.	1ncg_|W.C.	1syc_|W.C.	2sim_|W.C.
1cgjE|W.C.	1fem_|W.C.	1icn_|W.C.	1nchA|W.C.	1syd_|W.C.	2sns_|W.C.
1cgsL|1-112	1fen_|W.C.	1idbA|W.C.	1ncoA|W.C.	1syf_|W.C.	2snwA|W.C.
1cgsL|113-219	1fga_|W.C.	1ifhL|1-108	1nesE|W.C.	1syg_|W.C.	2sob_|W.C.

1cgt_|580-684	1fmd1|W.C.	1ifhL|109-211	1niaA|8-166	1tgb_|W.C.	2tbs_|W.C.
1cgt_|383-494	1fnc_|19-154	1igjA|1-107	1nibA|8-166	1tgc_|W.C.	2tga_|W.C.
1cgt_|495-579	1fnd_|19-154	1igjA|108-211	1nic_|8-166	1tgn_|W.C.	2tgd_|W.C.
1cgu_|383-494	1fnf_|1142-123	1igp_|W.C.	1nid_|8-166	1thaA|W.C.	2tgpZ|W.C.
1cgu_|495-579	1fod1|W.C.	1ihtL|W.C.	1nie_|8-166	1thcA|W.C.	2trm_|W.C.
1cgu_|580-684	1fpcL|W.C.	1iluA|W.C.	1nmaL|W.C.	1thrL|W.C.	2tsaA|W.C.
1cgv_|383-495	1frn_|19-154	1inc_|W.C.	1nmaN|W.C.	1thsL|W.C.	2tsbA|W.C.
1cgv_|496-581	1fveA|1-108	1ineL|2-109	1nn2_|W.C.	1thu_|W.C.	2tunA|W.C.
1cgv_|582-686	1fveA|109-214	1ineL|110-212	1nna_|W.C.	1thv_|W.C.	2vaaA|182-274
1cgw_|383-495	1gbaA|W.C.	1inv_|W.C.	1nol_|380-628	1tld_|W.C.	3app_|W.C.
1cgw_|496-581	1gbbA|W.C.	1inw_|W.C.	1nrnL|W.C.	1tlmA|W.C.	3bj1A|W.C.
1cgw_|582-686	1gbcA|W.C.	1inx_|W.C.	1nroL|W.C.	1tmbL|W.C.	3cysA|W.C.
1cgy_|383-495	1gbfA|W.C.	1iny_|W.C.	1nrqL|W.C.	1tmcB|W.C.	3er3E|W.C.
1cgy_|496-581	1gbhA|W.C.	1irp_|W.C.	1nrrL|W.C.	1tmf1|W.C.	3er5E|W.C.
1cgy_|582-686	1gbiA|W.C.	1ivb_|W.C.	1nrsL|W.C.	1tmtL|W.C.	3hatL|W.C.
1chg_|W.C.	1gbjA|W.C.	1ivc_|W.C.	1nsbA|W.C.	1tmuL|W.C.	3hudA|1-174
1choE|W.C.	1gbkA|W.C.	1ivd_|W.C.	1nsdA|W.C.	1tnfA|W.C.	3ptb_|W.C.
1chqD|W.C.	1gblA|W.C.	1ive_|W.C.	1ntp_|W.C.	1tng_|W.C.	4ape_|W.C.
1cil_|W.C.	1gbmA|W.C.	1ivf_|W.C.	1nzrA|W.C.	1tnh_|W.C.	4azuA|W.C.
1cin_|W.C.	1gcd_|W.C.	1ivg_|W.C.	1opbA|W.C.	1tni_|W.C.	4er1E|W.C.
1ckbA|W.C.	1gcs_|1-85	1ivpA|W.C.	1oxy_|380-627	1tnj_|W.C.	4htcL|W.C.
1cn1A|W.C.	1gfc_|W.C.	1ivqA|W.C.	1p01A|W.C.	1tnk_|W.C.	4pep_|W.C.
1cnc_|W.C.	1gfd_|W.C.	1jim_|W.C.	1p02A|W.C.	1tnl_|W.C.	4rcrH|36-248
1cneA|11-124	1ggcL|1-107	1kaa_|W.C.	1p03A|W.C.	1tnn_|W.C.	5cac_|W.C.
1cnf_|11-124	1ggcL|108-211	1kab_|W.C.	1p04A|W.C.	1tpaE|W.C.	5chaA|W.C.
1cng_|W.C.	1ggiL|1-107	1kda_|W.C.	1p05A|W.C.	1tpo_|W.C.	5cnaA|W.C.
1cnh_|W.C.	1ggiL|108-211	1kdb_|W.C.	1p06A|W.C.	1tpp_|W.C.	5er2E|W.C.
1cni_|W.C.	1ghaE|W.C.	1kdc_|W.C.	1p09A|W.C.	1tps_|W.C.	9lprA|W.C.
1cnj_|W.C.	1ghbE|W.C.	1knoA|1-108	1p10A|W.C.	1trmA|W.C.	12ca_|W.C.
1cnk_|W.C.	1glbF|W.C.	1knoA|109-214	1p11E|W.C.	1ttbA|W.C.	31bi_|W.C.
1cnw_|W.C.	1glcF|W.C.	1kraB|W.C.	1p12E|W.C.	1ttf_|W.C.	
1cny_|W.C.	1gldF|W.C.	1kraC|2-129	1piv1|W.C.	1ttg_|W.C.	
1cobA|W.C.	1gleF|W.C.	1krbB|W.C.	1pks_|W.C.	1tyn_|W.C.	
1conA|W.C.	1glh_|W.C.	1krbC|2-129	1pkt_|W.C.	1tyrA|W.C.	

608 α+β domains

1aarA|W.C.	1glv_|123-316	1l64_|W.C.	1mri_|W.C.	1rsm_|W.C.	4ltyA|W.C.
1acbI|W.C.	1gmqA|W.C.	1l65_|W.C.	1mrk_|W.C.	1rsnA|W.C.	4mdhA|155-333
1acmB|8-100	1gmrA|W.C.	1l66_|W.C.	1msc_|W.C.	1rtb_|W.C.	5ldh_|163-331
1aec_|W.C.	1gra_|364-478	1l67_|W.C.	1msdA|84-198	1rtnA|W.C.	6fdr_|W.C.
1afb1|105-221	1grb_|364-478	1l68_|W.C.	1msgA|W.C.	1rusA|3-137	6ldh_|161-329
1afd1|105-221	1gre_|364-478	1l69_|W.C.	1mshA|W.C.	1sbnI|W.C.	6lyt_|W.C.
1aha_|W.C.	1grf_|364-478	1l70_|W.C.	1mstA|W.C.	1sceA|W.C.	8atcB|8-100
1ahb_|W.C.	1grg_|364-478	1l72_|W.C.	1ndaA|358-484	1sesA|111-421	9ldb_|163-331
1ahc_|W.C.	1grl_|137-190	1l73_|W.C.	1ndc_|W.C.	1setA|111-421	9pap_|W.C.
1akl_|1-239	1hcsB|W.C.	1l74_|W.C.	1ndk_|W.C.	1shbA|W.C.	102l_|W.C.
1atnD|W.C.	1hctB|W.C.	1l75_|W.C.	1ndlA|W.C.	1sphA|W.C.	103l_|W.C.
1aybA|W.C.	1hdn_|W.C.	1l76_|W.C.	1ndpA|W.C.	1sprA|W.C.	104lA|W.C.
1aycA|W.C.	1hel_|W.C.	1l77_|W.C.	1nel_|1-141	1spsA|W.C.	107l_|W.C.
1ayd_|W.C.	1hem_|W.C.	1l79_|W.C.	1nhb_|W.C.	1srnA|W.C.	108l_|W.C.
1baoA|W.C.	1hen_|W.C.	1l80_|W.C.	1nhp_|322-447	1srp_|4-239	109l_|W.C.
1bdmA|155-332	1heo_|W.C.	1l81_|W.C.	1nhq_|322-447	1ssaA|W.C.	110l_|W.C.
1bgsA|W.C.	1hep_|W.C.	1l82_|W.C.	1nhr_|322-447	1ssbA|W.C.	111l_|W.C.

1bib_|64-270	1heq_|W.C.	1l83_|W.C.	1nhs_|322-447	1sscA|W.C.	112l_|W.C.
1bmsA|W.C.	1her_|W.C.	1l84_|W.C.	1nlkR|W.C.	1svr_|W.C.	114l_|W.C.
1bneA|W.C.	1hew_|W.C.	1l85_|W.C.	1nskR|W.C.	1synA|W.C.	115l_|W.C.
1bnfA|W.C.	1hhgA|1-181	1l86_|W.C.	1nsp_|W.C.	1tay|W.C.	118l_|W.C.
1bngA|W.C.	1hhhA|1-181	1l87_|W.C.	1pafA|W.C.	1tbeA|W.C.	119l_|W.C.
1bniA|W.C.	1hhiA|1-181	1l88_|W.C.	1pagA|W.C.	1tbpA|61-155	120l_|W.C.
1bnjA|W.C.	1hhjA|1-181	1l89_|W.C.	1pbb_|174-275	1tby_|W.C.	122l_|W.C.
1bnr_|W.C.	1hhkA|1-181	1l90_|W.C.	1pbc_|174-275	1tcs_|W.C.	123l_|W.C.
1bnsA|W.C.	1hnl_|W.C.	1l91_|W.C.	1pbd_|174-275	1tcy_|W.C.	125l_|W.C.
1brgA|W.C.	1htdA|W.C.	1l93_|W.C.	1pbf_|174-275	1tda_|W.C.	126l_|W.C.
1brhA|W.C.	1htlA|W.C.	1l94_|W.C.	1pdh_|174-275	1tdb_|W.C.	127l_|W.C.
1briA|W.C.	1hunA|W.C.	1l96_|W.C.	1pdy_|1-139	1tdc_|W.C.	128l_|W.C.
1brjA|W.C.	1hymA|W.C.	1l97A|W.C.	1pe6_|W.C.	1tdy_|W.C.	129l_|W.C.
1brkA|W.C.	1hyt_|1-155	1l99_|W.C.	1pfh_|W.C.	1tew_|W.C.	131l_|W.C.
1brsA|W.C.	1iaa_|W.C.	1laa_|W.C.	1pfmA|W.C.	1thl_|1-155	132l_|W.C.
1bsaA|W.C.	1iab_|W.C.	1lca_|W.C.	1pfnA|W.C.	1thy_|W.C.	133l_|W.C.
1bsbA|W.C.	1iac_|W.C.	1lcb_|W.C.	1pgb_|W.C.	1tla_|W.C.	134l_|W.C.
1bsdA|W.C.	1iad_|W.C.	1lce_|W.C.	1pgx_|W.C.	1tlpE|1-155	135l_|W.C.
1bseA|W.C.	1iae_|W.C.	1lcjA|W.C.	1pipA|W.C.	1tmc_|W.C.	137lA|W.C.
1cge_|W.C.	1ikl_|W.C.	1lcoA|10-97	1plr_|1-126	1trqA|W.C.	138l_|W.C.
1cgfA|W.C.	1ikm_|W.C.	1ldb_|163-331	1pnlA|W.C.	1tsdA|W.C.	139l_|W.C.
1cglA|W.C.	1isbA|83-192	1ldcA|10-97	1pnmA|W.C.	1tsw_|W.C.	140l_|W.C.
1ciqA|W.C.	1iscA|83-192	1lhh_|W.C.	1popA|W.C.	1tsx_|W.C.	141l_|W.C.
1cirA|W.C.	1ius_|174-275	1lhi_|W.C.	1ppd_|W.C.	1tsy_|W.C.	142l_|W.C.
1coaI|W.C.	1iut_|174-275	1lhj_|W.C.	1ppp_|W.C.	1tsz_|W.C.	143l_|W.C.
1comA|W.C.	1iuu_|174-275	1lhl_|W.C.	1ptoA|W.C.	1typA|359-487	144l_|W.C.
1coy_|319-450	1kraA|W.C.	1lhm_|W.C.	1ptoB|4-87	1tys_|W.C.	145l_|W.C.
1cpjA|W.C.	1krbA|W.C.	1lklA|W.C.	1pxa_|174-275	1tytA|359-487	147l_|W.C.
1csbA|W.C.	1krcA|W.C.	1lma_|W.C.	1pxb_|174-275	1ubq_|W.C.	148lE|W.C.
1cteA|W.C.	1l00_|W.C.	1lmc_|W.C.	1pxc_|174-275	1umsA|W.C.	149l_|W.C.
1cyo_|W.C.	1l02_|W.C.	1lmn_|W.C.	1raaB|1-100	1umtA|W.C.	150lA|W.C.
1cyu_|W.C.	1l03_|W.C.	1lmo_|W.C.	1rabB|1-100	1vfbC|W.C.	151l_|W.C.
1dktA|W.C.	1l04_|W.C.	1lmp_|W.C.	1racB|1-100	1vig_|W.C.	152l_|W.C.
1dob_|174-275	1l05_|W.C.	1lmt_|W.C.	1radB|1-100	1xra_|1-101	154l_|W.C.
1dod_|174-275	1l06_|W.C.	1lnaE|1-155	1raeB|1-100	1xrc_|1-101	155l_|W.C.
1doe_|174-275	1l07_|W.C.	1lnbE|1-155	1rafB|1-100	1xxbA|W.C.	156l_|W.C.
1doy_|W.C.	1l08_|W.C.	1lncE|1-155	1rahB|1-100	1xxcA|W.C.	157l_|W.C.
1dtp_|W.C.	1l09_|W.C.	1lndE|1-155	1raiB|1-100	1yam_|W.C.	158l_|W.C.
1dya_|W.C.	1l10_|W.C.	1lneE|1-155	1rar_|W.C.	1yan_|W.C.	159l_|W.C.
1dyb_|W.C.	1l11_|W.C.	1lnfE|1-155	1ras_|W.C.	1yao_|W.C.	160l_|W.C.
1dyc_|W.C.	1l12_|W.C.	1lpfA|349-472	1rbaA|5-137	1yap_|W.C.	161l_|W.C.
1dyd_|W.C.	1l13_|W.C.	1lra_|W.C.	1rbbA|W.C.	1yaq_|W.C.	162l_|W.C.
1dye_|W.C.	1l14_|W.C.	1lsa_|W.C.	1rbc_|W.C.	1ypaI|W.C.	164l_|W.C.
1dyf_|W.C.	1l16_|W.C.	1lsb_|W.C.	1rbd_|W.C.	1ypbI|W.C.	165l_|W.C.
1dyg_|W.C.	1l17_|W.C.	1lsc_|W.C.	1rbe_|W.C.	1ypcI|W.C.	166l_|W.C.
1e8l_|W.C.	1l18_|W.C.	1lsd_|W.C.	1rbf_|W.C.	2aadA|W.C.	167lA|W.C.
1ebgA|1-141	1l19_|W.C.	1lse_|W.C.	1rbg_|W.C.	2aae_|W.C.	168lA|W.C.
1ebhA|1-141	1l20_|W.C.	1lsf_|W.C.	1rbw_|W.C.	2aas_|W.C.	169lA|W.C.
1egl_|W.C.	1l21_|W.C.	1lsg_|W.C.	1rbx_|W.C.	2acg_|W.C.	170l_|W.C.
1egpA|W.C.	1l22_|W.C.	1lsm_|W.C.	1rca_|W.C.	2atcB|1-100	171l_|W.C.
1els_|1-141	1l23_|W.C.	1lsn_|W.C.	1rck_|W.C.	2baa_|W.C.	172l_|W.C.
1emd_|146-312	1l25_|W.C.	1lsp_|W.C.	1rcl_|W.C.	2bbqA|W.C.	173l_|W.C.
1esp_|1-156	1l26_|W.C.	1lsy_|W.C.	1rdi1|W.C.	2chtA|W.C.	174lA|W.C.

1fcbA|1-97	1l27_|W.C.	1lsz_|W.C.	1rdj1|W.C.	2ci2I|W.C.	175lA|W.C.
1fccC|W.C.	1l28_|W.C.	1ltaA|W.C.	1rdk1|W.C.	2fke_|W.C.	177l_|W.C.
1fd2_|W.C.	1l29_|W.C.	1ltbA|W.C.	1rdl1|W.C.	2hpr_|W.C.	178l_|W.C.
1fda_|W.C.	1l30_|W.C.	1ltgA|W.C.	1rdm1|W.C.	2iffY|W.C.	179l_|W.C.
1fdb_|W.C.	1l31_|W.C.	1lthR|150-319	1rdn1|W.C.	2igg_|W.C.	180lA|W.C.
1fdd_|W.C.	1l32_|W.C.	1lttA|W.C.	1rdo1|W.C.	2igh_|W.C.	181l_|W.C.
1fdn_|W.C.	1l33_|W.C.	1lvl_|336-458	1rds_|W.C.	2l78_|W.C.	182l_|W.C.
1fer_|W.C.	1l34_|W.C.	1lyd_|W.C.	1rga_|W.C.	2lz2_|W.C.	183l_|W.C.
1fkb_|W.C.	1l35_|W.C.	1lye_|W.C.	1rgcA|W.C.	2lzt_|W.C.	184l_|W.C.
1fkf_|W.C.	1l36_|W.C.	1lyf_|W.C.	1rgl_|W.C.	2mhaA|1-181	185l_|W.C.
1fkg_|W.C.	1l37_|W.C.	1lyh_|W.C.	1rha_|W.C.	2nckR|W.C.	186l_|W.C.
1fkh_|W.C.	1l38_|W.C.	1lyi_|W.C.	1rhb_|W.C.	2phh_|174-275	187l_|W.C.
1fkiA|W.C.	1l39_|W.C.	1lyj_|W.C.	1rlcL|22-147	2pleA|W.C.	188l_|W.C.
1fkj_|W.C.	1l40_|W.C.	1lysA|W.C.	1rlcS|W.C.	2pnb_|W.C.	190l_|W.C.
1fkk_|W.C.	1l42_|W.C.	1lz1_|W.C.	1rldA|22-147	2rlnE|W.C.	191l_|W.C.
1fkl_|W.C.	1l43_|W.C.	1lz4_|W.C.	1rldS|W.C.	2rns_|W.C.	192l_|W.C.
1fks_|W.C.	1l44_|W.C.	1lza_|W.C.	1rls_|W.C.	2rusA|2-137	194l_|W.C.
1fkt_|W.C.	1l45_|W.C.	1lzb_|W.C.	1rn4_|W.C.	2sarA|W.C.	195l_|W.C.
1fmp_|W.C.	1l46_|W.C.	1lzc_|W.C.	1rnc_|W.C.	2secI|W.C.	196l_|W.C.
1frh_|W.C.	1l47_|W.C.	1lzd_|W.C.	1rnd_|W.C.	2tcl_|W.C.	197l_|W.C.
1fri_|W.C.	1l48_|W.C.	1lzg_|W.C.	1rnlA|W.C.	2tdd_|W.C.	198l_|W.C.
1frk_|W.C.	1l50_|W.C.	1lzsA|W.C.	1rnmE|W.C.	2tdm_|W.C.	199l_|W.C.
1frl_|W.C.	1l52_|W.C.	1lzy_|W.C.	1rno_|W.C.	2tecI|W.C.	200l_|W.C.
1frm_|W.C.	1l54_|W.C.	1mdr_|3-132	1rnq_|W.C.	2tscA|W.C.	201lA|W.C.
1frx_|W.C.	1l55_|W.C.	1mdtA|1-187	1rnu_|W.C.	2vaaA|W.C.	216lA|W.C.
1fut_|W.C.	1l56_|W.C.	1mit_|W.C.	1rnv_|W.C.	3ci2_|W.C.	217l_|W.C.
1fxaA|W.C.	1l57_|W.C.	1mlcE|W.C.	1rob_|W.C.	3dni_|W.C.	221l_|W.C.
1gaeO|149-312	1l58_|W.C.	1mmpA|W.C.	1rpf_|W.C.	3mdsA|93-203	224l_|W.C.
1gb1_|W.C.	1l59_|W.C.	1mmr_|W.C.	1rpg_|W.C.	3monA|W.C.	
1gesA|336-450	1l60_|W.C.	1mns_|3-132	1rph_|W.C.	3rn3_|W.C.	
1getA|336-450	1l61_|W.C.	1mom_|W.C.	1rcsA|9-147	3ssi_|W.C.	
1geuA|336-450	1l62_|W.C.	1mrh_|W.C.	1rscM|W.C.	4gr1_|364-478	

509 α/β domains

1aaw_|W.C.	1cec_|W.C.	1ffa_|W.C.	1lap_|160-484	1raeA|1-150	1ula_|W.C.
1aba_|W.C.	1cen_|W.C.	1ffb_|W.C.	1lav_|W.C.	1rafA|1-150	1vlzA|W.C.
1abbA|W.C.	1cey_|W.C.	1ffc_|W.C.	1law_|W.C.	1rahA|1-150	1vruA|430-539
1abf_|W.C.	1cgt_|1-382	1ffd_|W.C.	1lbs_|W.C.	1raiA|1-150	1vse_|W.C.
1ace_|W.C.	1cgu_|1-382	1ffe_|W.C.	1lbt_|W.C.	1rbaA|138-441	1vsf_|W.C.
1acj_|W.C.	1cgv_|1-382	1flv_|W.C.	1lcf_|1-334	1rbr_|W.C.	1whsA|W.C.
1acl_|W.C.	1cgw_|1-382	1fnc_|155-314	1lcoA|98-511	1rbs_|W.C.	1wsyA|W.C.
1acmA|1-150	1cgy_|1-382	1fnd_|155-314	1ldb_|15-162	1rbt_|W.C.	1wsyB|W.C.
1adbA|175-324	1chn_|W.C.	1frn_|155-314	1ldcA|98-511	1rbu_|W.C.	1xab_|W.C.
1adcA|175-324	1cia_|W.C.	1fx1_|W.C.	1lfh_|1-334	1rbv_|W.C.	1xad_|W.C.
1adf_|175-324	1cne_|125-270	1gaeO|0-148	1lfi_|1-334	1rda_|W.C.	1xib_|W.C.
1adg_|175-324	1cnf_|125-270	1gcg_|W.C.	1lgbC|W.C.	1rdb_|W.C.	1xic_|W.C.
1ads_|W.C.	1coy_|4-318	1gdd_|9-60	1lpaB|1-336	1rdc_|W.C.	1xid_|W.C.
1agp_|W.C.	1cps_|W.C.	1gesA|3-146	1lpfA|1-158	1rhd_|1-149	1xie_|W.C.
1aheA|W.C.	1cpy_|W.C.	1getA|3-146	1lpm_|W.C.	1rlcL|148-467	1xif_|W.C.
1ahfA|W.C.	1crp_|W.C.	1geuA|3-146	1lpn_|W.C.	1rldA|148-467	1xig_|W.C.
1ahgA|W.C.	1crq_|W.C.	1gfi_|33-60	1lpo_|W.C.	1rnh_|W.C.	1xii_|W.C.
1ahxA|W.C.	1ctu_|1-150	1gil_|34-60	1lpp_|W.C.	1rpt_|W.C.	1xij_|W.C.
1ahyA|W.C.	1cxe_|W.C.	1glbG|4-253	1lps_|W.C.	1rscA|148-475	1xlaA|W.C.

1aiaA|W.C.	1cxf_|1-382	1glcG|4-253	1lthR|7-149	1rthA|430-543	1xlcA|W.C.
1aibA|W.C.	1cxg_|1-382	1gldG|4-253	1lvl_|1-150	1rtiA|430-543	1xldA|W.C.
1aicA|W.C.	1cxh_|1-382	1gleG|4-253	1map_|W.C.	1rtjA|430-543	1xleA|W.C.
1ajaA|W.C.	1cxi_|1-382	1glg_|W.C.	1maq_|W.C.	1s01_|W.C.	1xlfA|W.C.
1ajcA|W.C.	1cye_|1-382	1glpA|1-78	1mdiA|W.C.	1s02_|W.C.	1xlgA|W.C.
1ajdA|W.C.	1dbp_|W.C.	1glv_|1-122	1mdjA|W.C.	1sbc_|W.C.	1xlhA|W.C.
1akaA|W.C.	1dbs_|W.C.	1gne_|1-79	1mdkA|W.C.	1sbh_|W.C.	1xliA|W.C.
1akcA|W.C.	1ddrA|W.C.	1gnp_|W.C.	1mdp1|W.C.	1sbi_|W.C.	1xljA|W.C.
1alhA|W.C.	1ddsA|W.C.	1gnq_|W.C.	1mdq_|W.C.	1sbnE|W.C.	1xlkA|W.C.
1aljA|W.C.	1dgd_|W.C.	1gnr_|W.C.	1mdr_|133-359	1sca_|W.C.	1xllA|W.C.
1alkA|W.C.	1dge_|W.C.	1goa_|W.C.	1mns_|133-359	1scb_|W.C.	1xyaA|W.C.
1ama_|W.C.	1dhiA|W.C.	1goc_|W.C.	1mpc_|W.C.	1scd_|W.C.	1xybA|W.C.
1ami_|2-528	1dhjA|W.C.	1gpaA|W.C.	1mpd_|W.C.	1scnE|W.C.	1xycA|W.C.
1ami_|529-754	1didA|W.C.	1gpy_|W.C.	1mssA|W.C.	1selA|W.C.	1xylA|W.C.
1amj_|2-528	1dieA|W.C.	1gra_|18-165	1ndaA|4-169	1st2_|W.C.	1xymA|W.C.
1amj_|529-754	1dirA|W.C.	1grb_|18-165	1nel_|142-436	1sto_|W.C.	1ymuA|W.C.
1amn_|W.C.	1dis_|W.C.	1grcA|W.C.	1nga_|4-188	1sub_|W.C.	1ymv_|W.C.
1amq_|W.C.	1diu_|W.C.	1gre_|18-165	1ngb_|4-188	1suc_|W.C.	1ypiA|W.C.
1amr_|W.C.	1dka_|W.C.	1grf_|18-165	1ngc_|4-188	1sud_|W.C.	1yptA|W.C.
1ams_|W.C.	1dlr_|W.C.	1grg_|18-165	1ngd_|4-188	1tag_|27-56	2acq_|W.C.
1aniA|W.C.	1dls_|W.C.	1grl_|191-375	1nge_|4-188	1tarA|W.C.	2acr_|W.C.
1anjA|W.C.	1dmb_|W.C.	1gro_|W.C.	1ngf_|3-188	1tasA|W.C.	2acu_|W.C.
1ankA|W.C.	1dob_|1-173	1grp_|W.C.	1ngg_|3-188	1tatA|W.C.	2ada_|W.C.
1apb_|W.C.	1dod_|1-173	1grx_|W.C.	1ngi_|4-188	1tcbA|W.C.	2anhA|W.C.
1argA|W.C.	1doe_|1-173	1gsdA|2-80	1ngj_|3-188	1tccA|W.C.	2bgt_|W.C.
1arhA|W.C.	1dot_|1-334	1gsfA|2-80	1nhp_|1-119	1tdf_|1-118	2che_|W.C.
1ariA|W.C.	1dpb_|W.C.	1gtb_|1-80	1nhq_|1-119	1tdrA|W.C.	2ctc_|W.C.
1ars_|W.C.	1dpc_|W.C.	1guhA|2-80	1nhr_|1-119	1tho_|W.C.	2cut_|W.C.
1asa_|W.C.	1dpd_|W.C.	1gylA|W.C.	1nhs_|1-119	1tkaA|3-337	2dhc_|W.C.
1asb_|W.C.	1dr1_|W.C.	1hdxA|175-324	1nis_|2-528	1tkaA|535-680	2dhd_|W.C.
1asc_|W.C.	1dr2_|W.C.	1hdyA|175-324	1nis_|529-754	1tkbA|3-337	2dhe_|W.C.
1asd_|W.C.	1dr3_|W.C.	1hdzA|175-324	1nit_|2-528	1tkbA|535-680	2eda_|W.C.
1asf_|W.C.	1dr4_|W.C.	1hex_|W.C.	1nit_|529-754	1tkcA3-337	2edc_|W.C.
1asg_|W.C.	1dr5_|W.C.	1hey_|W.C.	1nnt_|W.C.	1tkcA|535-680	2glrA|1-78
1aslA|W.C.	1dr6_|W.C.	1hldA|175-324	1olaA|W.C.	1tndA|27-56	2hnp_|W.C.
1asmA|W.C.	1dr7_|W.C.	1hmvA|430-554	1olcA|W.C.	1tpb1|W.C.	2hsdA|W.C.
1asnA|W.C.	1draA|W.C.	1hnbA|1-84	1omp_|W.C.	1tpc1|W.C.	2lao_|W.C.
1asu_|W.C.	1drbA|W.C.	1hncA|1-84	1orb_|1-149	1tpdA|W.C.	2nadA|1-147
1asv_|W.C.	1drf_|W.C.	1hniA|430-556	1oya|W.C.	1tpe_|W.C.	2nadA|148-335
1asw_|W.C.	1drh_|W.C.	1horA|W.C.	1oyc_|W.C.	1tpfA|W.C.	2oxiA|175-324
1atnA|0-146	1drj_|W.C.	1hotA|W.C.	1pbb_|1-173	1tpuA|W.C.	2phh_|1-173
1atr_|2-188	1drk_|W.C.	1hqaA|W.C.	1pbc_|1-173	1tpvA|W.C.	2pkc_|W.C.
1ats_|2-188	1dsn_|W.C.	1hrhA|W.C.	1pbd_|1-173	1tpwA|W.C.	2pri_|W.C.
1bap_|W.C.	1dvrA|W.C.	1htbA|175-324	1pbf_|1-173	1trb_|1-118	2prj_|W.C.
1bcmA|257-480	1dyhA|W.C.	1hvm_|W.C.	1pbp_|W.C.	1trdA|W.C.	2rusA|138-457
1bcrA|W.C.	1dyiA|W.C.	1hvq_|W.C.	1pdh_|1-173	1trh_|W.C.	2secE|W.C.
1bcsA|W.C.	1dyjA|W.C.	1idd_|W.C.	1pdy_|140-433	1tri_|W.C.	2ts1_|W.C.
1bdmA|0-154	1eaa_|W.C.	1ide_|W.C.	1pekE|W.C.	1trs_|W.C.	2tecE|W.C.
1bgsE|W.C.	1eab_|W.C.	1idm_|W.C.	1pgn_|1-176	1tru_|W.C.	2tir_|W.C.
1bllE|1-159	1eac_|W.C.	1ika_|W.C.	1pgo_|1-176	1trv_|W.C.	3drcA|W.C.
1bllE|160-484	1ead_|W.C.	1ikb_|W.C.	1pgp_|1-176	1trw_|W.C.	3hsc_|3-188
1bpm_|1-159	1eae_|W.C.	1ipd_|W.C.	1pgq_|1-176	1tsiA|W.C.	3hudA|175-324
1bpm_|160-484	1ebgA|142-436	1ius_|1-173	1phh_|1-173	1tti_|W.C.	3hvtA|430-556

1bpn_|1-159	1ebhA|142-436	1iut_|1-173	1plj_|W.C.	1ttj_|W.C.	3sc2A|W.C.
1bpn_|160-484	1edb_|W.C.	1iuu_|1-173	1pll_|W.C.	1ttpA|W.C.	4gr1_|18-165
1brsD|W.C.	1edd_|W.C.	1kraC|130-422	1pnt_|W.C.	1ttpB|W.C.	4mdhA|1-154
1btb_|W.C.	1ede_|W.C.	1krbC|130-422	1poxA|9-182	1ttqA|W.C.	4q21_|W.C.
1btc_|W.C.	1ego_|W.C.	1krcC|130-422	1poxA|183-365	1ttqB|W.C.	5abp_|W.C.
1bvh_|W.C.	1egr_|W.C.	1lafE|W.C.	1ptk_|W.C.	1tyaE|1-217	5ldh_|1-162
1bya_|W.C.	1els_|142-436	1lagE|W.C.	1pxa_|1-173	1tybE|1-217	6ldh_|1-160
1byc_|W.C.	1emd_|1-145	1lahE|W.C.	1pxb_|1-173	1tyc_|1-217	6q21A|W.C.
1byd_|W.C.	1enz_|W.C.	1lam_|1-159	1pxc_|1-173	1tydE|1-217	8atcA|1-150
1cbx_|W.C.	1esd_|W.C.	1lam_|160-484	1raaA|1-150	1typA|1-169	9icd_|W.C.
1cddA|W.C.	1ese_|W.C.	1lanA|1-159	1rabA|1-150	1tytA|1-169	9ldbA|1-162
1cde_|W.C.	1etu_|W.C.	1lanA|160-484	1racA|1-150	1udg_|W.C.	121p_|W.C.
1cdoA|176-324	1fcbA|98-511	1lap_|1-159	1radA|1-150	1uky_|W.C.	

158 σ domains

1aalA|W.C.	1coe_|W.C.	1hfi_|W.C.	1nag_|W.C.	1radB|101-153	2crt_|W.C.
1acmB|101-153	1crf_|W.C.	1hic_|W.C.	1ncpC|W.C.	1raeB|101-153	2cthA|W.C.
1agg_|W.C.	1crn_|W.C.	1hiqA|W.C.	1neh_|W.C.	1rafB|101-153	2cwgA|1-52
1aphA|W.C.	1cti_|W.C.	1hisA|W.C.	1nrb_|W.C.	1rahB|101-153	2cym_|W.C.
1are_|W.C.	1cvo_|W.C.	1hitA|W.C.	1nxb_|W.C.	1raiB|101-153	2gda_|W.C.
1arf_|W.C.	1cxn_|W.C.	1hlsA|W.C.	1oav_|W.C.	1rgd_|W.C.	2hir_|W.C.
1atb_|W.C.	1cxo_|W.C.	1hrf_|W.C.	1oaw_|W.C.	1sgqI|W.C.	2hiuA|W.C.
1atd_|W.C.	1den_|W.C.	1hrpA|W.C.	1omb_|W.C.	1sgrI|W.C.	2hpqP|W.C.
1ate_|W.C.	1dmd_|W.C.	1hrq_|W.C.	1omt_|W.C.	1shi_|W.C.	2kaiI|W.C.
1bbi_|W.C.	1dme_|W.C.	1hrr_|W.C.	1omu_|W.C.	1tch_|W.C.	2let_|W.C.
1bonA|W.C.	1dmf_|W.C.	1hrtI|W.C.	1paa_|W.C.	1tcj_|W.C.	2nbtA|W.C.
1bphA|W.C.	1dphA|W.C.	1igl_|W.C.	1pcn_|1-44	1tck_|W.C.	2pf1_|36-65
1brcI|W.C.	1edp_|W.C.	1ihtI|W.C.	1pco_|1-44	1tcp_|W.C.	2ptcI|W.C.
1btgA|W.C.	1ehs_|W.C.	1irn_|W.C.	1pcp_|1-53	1tfg_|W.C.	2spt_|1-65
1bti_|W.C.	1epg_|W.C.	1iva_|W.C.	1pi2_|W.C.	1tmr_|W.C.	2tciA|W.C.
1cad_|W.C.	1eph_|W.C.	1izaA|W.C.	1pih_|W.C.	1tpaI|W.C.	2tgpI|W.C.
1cbn_|W.C.	1epi_|W.C.	1izbA|W.C.	1pij_|W.C.	1tpm_|W.C.	2wgcA|1-52
1ccf_|W.C.	1epj_|W.C.	1ldr_|W.C.	1pit_|W.C.	1tpn_|W.C.	3cyr_|W.C.
1ccm_|W.C.	1era_|W.C.	1lpaA|6-44	1pk2_|W.C.	1tur_|W.C.	3mthA|W.C.
1ccn_|W.C.	1etm_|W.C.	1maeL|W.C.	1pkr_|W.C.	1tus_|W.C.	4htcI|W.C.
1cdq_|W.C.	1etn_|W.C.	1mafL|W.C.	1pmkA|W.C.	1tylA|W.C.	5pti_|W.C.
1cdr_|W.C.	1fan_|W.C.	1mdaL|W.C.	1pmlA|W.C.	1tymA|W.C.	8atcB|101-153
1cds_|W.C.	1fra_|W.C.	1med_|W.C.	1prhA|33-73	1vnb_|W.C.	9wgaA|1-52
1cebA|W.C.	1fsc_|W.C.	1mhiA|W.C.	1ptr_|W.C.	1zrp_|W.C.	
1cgiI|W.C.	1gdc_|W.C.	1mhjA|W.C.	1raaB|101-153	2abxA|W.C.	
1cgjI|W.C.	1hcc_|W.C.	1mpjA|W.C.	1rabB|101-153	2atcB|101-152	
1choI|W.C.	1hcp_|W.C.	1mrt_|W.C.	1racB|101-153	2cco_|W.C.	

46 μ domains

1antI|W.C.	1bpd_|92-335	1fbfA|W.C.	1fprA|W.C.	1imeA|W.C.	2bpc_|W.C.
1apmE|W.C.	1bpe_|92-335	1fbgA|W.C.	1har_|W.C.	1imf_|W.C.	2cah_|W.C.
1atpE|W.C.	1ckjA|W.C.	1fbhA|W.C.	1hmvA|1-429	1mblA|W.C.	2glsA|W.C.
1blc_|W.C.	1cmkE|W.C.	1fpbA|W.C.	1hniA|1-429	1pioA|W.C.	2lgsA|W.C.
1blh_|W.C.	1ctpE|W.C.	1fpdA|W.C.	1imaA|W.C.	1rthA|2-429	3hvtA|2-429
1blp_|W.C.	1fbcA|W.C.	1fpeA|W.C.	1imbA|W.C.	1rtiA|2-429	3mdeA|11-241
1blsA|W.C.	1fbdA|W.C.	1fpfA|W.C.	1imcA|W.C.	1rtjA|2-429	
1bpb_|W.C.	1fbeA|W.C.	1fpgA|W.C.	1imdA|W.C.	1vruA|3-429	

20 ρ domains

1amb_|W.C.	1bhb_|W.C.	1btt_|W.C.	1kb8_|W.C.	1pak_|W.C.	1tiv_|W.C.
1amc_|W.C.	1btr_|W.C.	1dtc_|W.C.	1nil_|W.C.	1pao_|W.C.	1tos_|W.C.
1bct_|W.C.	1bts_|W.C.	1gnb_|W.C.	1nim_|W.C.	1rpb_|W.C.	1tvt_|W.C.
1wfaA|W.C.	1xy2_|W.C.				

Table A.8 The 277 Protein Domains.

70 all-α domains					
1hbiA|W.C.	1sctA|W.C.	1ytc_|W.C.	1yea_|W.C.	1yeb_|W.C.	1csc_|W.C.
2pccB|W.C.	1fhb_|W.C.	1cih_|W.C.	1cie_|W.C.	1csu_|W.C.	1troA|W.C.
1crj_|W.C.	1csw_|W.C.	1csx_|W.C.	1chi_|W.C.	1cig_|W.C.	5cscsA|W.C.
1crh_|W.C.	1raq_|W.C.	1ctz_|W.C.	1chj_|W.C.	1cif_|W.C.	3wrp_|W.C.
1csv_|W.C.	1crg_|W.C.	1chh_|W.C.	1rap_|W.C.	1hddC|W.C.	1phb_|W.C.
1dprA|65-136	1tnt_|W.C.	1bbl_|W.C.	1erc_|W.C.	1aca_|W.C.	1trrA|W.C.
1vasA|W.C.	1enk_|W.C.	1eni_|W.C.	1lynA|W.C.	1hme_|W.C.	3fisA|W.C.
1hsm_|W.C.	1gnc_|W.C.	1rprA|W.C.	1rpo_|W.C.	1pou_|W.C.	1grl_|6-136
1cdn_|W.C.	1bod_|W.C.	1boc_|W.C.	1arqA|W.C.	1mykA|W.C.	1fipA|W.C.
1mylA|W.C.	1bpd_|W.C.	1olhA|W.C.	1pesA|W.C.	1hns_|W.C.	1afb1|73-104
1tag_|57-177	4ts1A|228-319	1tyc_|228-319	1lgaA|W.C.	1oxy_|1-379	1csi_|W.C.
1nol_|1-379	1pgn_|177-473	2utgA|W.C.	3gly_|W.C.		
61 all-β domains					
1mdtA|381-535	1cgt_|580-684	1cxe_|582-686	1aaj_|W.C.	1mdaA|W.C.	1gog_|151-537
1gcs_|W.C.	1pnf_|1-140	1png_|5-140	1gog_|1-150	1tnfA|W.C.	1azm_|W.C.
2tunA|W.C.	1thv_|W.C.	1thu_|W.C.	2ctvA|W.C.	1apnA|W.C.	1kraC|2-129
2cna_|W.C.	1bib_|271-317	1ltaD|W.C.	1bfb_|W.C.	1fga_|W.C.	1cgt_|383-494
2bfh_|W.C.	1bfg_|W.C.	1bas_|W.C.	1fnd_|19-154	1frn_|19-154	1bzm|W.C.
1arc_|W.C.	1bcmA|481-560	1hpxA|W.C.	1hivA|W.C.	1hshA|W.C.	1cxe_|383-495
1cpiA|W.C.	1hvrA|W.C.	1hvc_|W.C.	4phvA|W.C.	1hefE|W.C.	1huh_|W.C.
1aaqA|W.C.	1hvsA|W.C.	1gtsA|339-547	1hbp_|W.C.	1fen_|W.C.	1hug_|W.C.
1erb_|W.C.	1slfB |W.C.	1srgA|W.C.	1srjA|W.C.	1ptsA|W.C.	1akl_|247-470
1sleB|W.C.	1cyhA|W.C.	1mikA|W.C.	3cysA|W.C.	2sim_|W.C.	1crm_|W.C.
1hpcS|W.C.					
81 α+β domains					
1cgt_|1-382	1cxe_|1-382	1cxf_|1-382	1cgv_|1-382	1cgw_|1-382	2bgt_|W.C.
1cgy_|1-382	1cgx_|1-382	1cgu_|1-382	1btb_|W.C.	1brsD|W.C.	1ctu_|1-150
1bgsE|W.C.	1fnd_|155-314	1frn_|155-314	4ts1A|1-217	1tyc_|1-217	1wsyB|W.C.
1tydE|1-217	1tybE|1-217	1tyaE|1-217	1cdoA|176-324	1hldA|175-324	1drk_|W.C.
1horA|W.C.	2secE|W.C.	1scnE|W.C.	1selA|W.C.	1cia_|W.C.	1orb_|1-149
1pnt_|W.C.	2hnp_|W.C.	1trx_|W.C.	2tir_|W.C.	1tho_|W.C.	1dbp_|W.C.
1tkbA|535-680	1lam_|1-159	1bllE|1-159	1gdtA|1-140	3hsc_|3-188	1rhd_|1-149
1ngi_|4-188	1ngb_|4-188	1nga_|4-188	1ngg_|3-188	1ngh_|4-188	1drj_|W.C.
1atr_|2-188	1cde_|W.C.	1grcA|W.C.	1cddA|W.C.	1mhtA|W.C.	5acn_|1-528
1ama_|W.C.	1akaA|W.C.	1ula_|W.C.	1amn_|W.C.	1acj_|W.C.	1olcA|W.C.
1acl_|W.C.	2ctc_|W.C.	5cpa_|W.C.	1dr1_|W.C.	2anhA|W.C.	1ttqB|W.C.
1hgaA|W.C.	1alkA|W.C.	1ajaA|W.C.	1ajdA|W.C.	1anjA|W.C.	1acmA|1-150
1aljA|W.C.	1aniA|W.C.	1alhA|W.C.	1ajcA|W.C.	1xab_|W.C.	8atcA|1-150
1ipd_|W.C.	1idm_|W.C.	1raiA|1-150			
65 α/β domains					
1fut_|W.C.	2baa_|W.C.	1aec_|W.C.	2rat_|W.C.	2rns_|W.C.	1tsw_|W.C.
1ras_|W.C.	1sscA|W.C.	1ssbA|W.C.	1ssa_|W.C.	1rbd_|W.C.	1ltaA|W.C.
1kraA|W.C.	1pgx_|W.C.	1pgb_|W.C.	1igc_|W.C.	1fccC|W.C.	1lttA|W.C.
2igg_|W.C.	2igh_|W.C.	1coy_|319-450	3monA|W.C.	1frtA|1-178	1ltgA|W.C.
1fkj_|W.C.	1fkl_|W.C.	2secI|W.C.	1egpA|W.C.	2tecI|W.C.	1htlA|W.C.
1egl_|W.C.	1sbnI|W.C.	1sibI|W.C.	3mdsA|93-203	1vig_|W.C.	1mrk_|W.C.
1mns_|3-132	1grl_|137-190	1rldS|W.C.	1comA|W.C.	1gaeO|149-312	1glv_|123-316

1mstA|W.C.	1bmsA|W.C.	1msc_|W.C.	1grb_|364-478	1lklA|W.C.	3dni_|W.C.
1lcjA|W.C.	1lckA|117-226	1sphA|W.C.	2hpr_|W.C.	1sceA|W.C.	1dnkA|W.C.
1setA|111-421	2tscA|W.C.	1tsdA|W.C.	2bbqA|W.C.	1tsy_|W.C.	4dmhA|155-333
1xrc_|1-101	1tsx_|W.C.	1tys_|W.C.	3b5c_|W.C.	1tbpA|61-155	

Table A.9 The 498 Protein Domains.

107 all-α domains					
1hbiA|W.C.	1sctA|W.C.	1ytc_|W.C.	1yea_|W.C.	1yeb_|W.C.	1phe_|W.C.
2pccB|W.C.	1fhb_|W.C.	1cih_|W.C.	1cie_|W.C.	1csu-|W.C.	1troA|W.C.
1crj_|W.C.	1csw_|W.C.	1csx_|W.C.	1chi_|W.C.	1cig_|W.C.	1afa1|73-104
1crh_|W.C.	1raq_|W.C.	1ctz_|W.C.	1chj_|W.C.	1cif_|W.C.	1cp4_|W.C.
1csv_|W.C.	1crg_|W.C.	1chh_|W.C.	1rap_|W.C.	1hddC|W.C.	3wrp_|W.C.
1dprA|65-136	1tnt_|W.C.	1bbl_|W.C.	1erc_|W.C.	1aca_|W.C.	1afd1|73-104
1vasA|W.C.	1enk_|W.C.	1eni_|W.C.	1lynA|W.C.	1hme_|W.C.	1noo_|W.C.
1hmf_|W.C.	1hsm_|W.C.	1nhn_|W.C.	1gnc_|W.C.	1rprA|W.C.	1trrA|W.C.
1rpo_|W.C.	1pou_|W.C.	1cdn_|W.C.	1bod_|W.C.	1boc_|W.C.	3fisA|W.C.
2bca_|W.C.	1clb_|W.C.	1arqA|W.C.	1arrA|W.C.	1mykA|W.C.	1grl_|6-316
1mylA|W.C.	1bpd_|9-91	2bpgA|9-91	1olhA|W.C.	1pesA|W.C.	1fipA|W.C.
1petA|W.C.	1saeA|W.C.	1safA|W.C.	1sagA|W.C.	1sahA|W.C.	1afb1|73-104
1saiA|W.C.	1sajA|W.C.	1sakA|W.C.	1salA|W.C.	1hns_|W.C.	1phf_|W.C.
1tag_|57-177	1tndA|57-177	4ts1A|228-319	1tyc_|228-319	1tydE|228-319	1phg_|W.C.
1tybE|228-319	1tyaE|228-319	1lgaA|W.C.	1oxy_|1-379	1nol_|1-379	1phd_|W.C.
1pgn_|177-473	1pgo_|177-473	1pgp_|177-473	1pgq_|177-473	2utgA|W.C.	1pha_|W.C.
3gly_|W.C.	1dog_|W.C.	1agm_|W.C.	1csi_|W.C.	1css_|W.C.	2cpp_|W.C.
1csr_|W.C.	1csc_|W.C.	5cts_|W.C.	5cscsA|W.C.	1phb_|W.C.	
126 all-β domains					
1mdtA|381-535	1cgt_|580-684	1cxe_|582-686	1cxi_|582-686	1cxf_|582-686	1krcC|2-129
1cvg_|582-686	1cgw_|582-686	1cgy_|582-686	1cgx_|582-686	1aaj_|W.C.	1hug_|W.C.
1aan_|W.C.	2mtaA|W.C.	1mdaA|W.C.	1gcs_|1-85	1pnf_|1-140	1huh_|W.C.
1png_|5-140	1gog_|1-150	1goh_|1-150	1tnfA|W.C.	2tunA|W.C.	1crm_|W.C.
1thv_|W.C.	1thu_|W.C.	2ctvA|W.C.	1scr_|W.C.	1conA|W.C.	1akl_|247-470
5cnaA|W.C.	1apnA|W.C.	2cna_|W.C.	1cn1A|W.C.	1bib_|271-317	1azm_|W.C.
1ltaD|W.C.	1lttD|W.C.	1ltgD|W.C.	1ltbD|W.C.	1htlD|W.C.	1hpcA|W.C.
1bfb_|W.C.	1bfc_|W.C.	1fga_|W.C.	2bfh_|W.C.	1bfg_|W.C.	1bzm_|W.C.
1bas_|W.C.	1fnd_|19-154	1fnc_|19-154	1frn_|19-154	1arc_|W.C.	1kraC|2-129
1bcmA|481-560	1hpxA|W.C.	1hihA|W.C.	1hvjA|W.C.	1hvkA|W.C.	1czm_|W.C.
1hivA|W.C.	1hpvA|W.C.	1hsgA|W.C.	1hshA|W.C.	1hvlA|W.C.	1krbC|2-129
1cpiA|W.C.	1hvrA|W.C.	1htgA|W.C.	1hvc_|W.C.	4phvA|W.C.	1cxf_|383-495
1hosA|W.C.	1sbgA|W.C.	1hhp_|W.C.	5hvpA|W.C.	1hbvA|W.C.	1cgu_|383-494
1hefE|W.C.	1hpsA|W.C.	1hsiA|W.C.	1hegE|W.C.	1aaqA|W.C.	1cxh_|383-495
1htfA|W.C.	1hteA|W.C.	3hvp_|W.C.	3phv_|W.C.	1hvsA|W.C.	1cgx_|383-495
1gtsA|339-547	1hbp_|W.C.	1fen_|W.C.	1erb_|W.C.	1fel_|W.C.	1cxg_|383-495
1fem_|W.C.	1slfB|W.C.	1srgA|W.C.	1sreA|W.C.	1srjA|W.C.	1cgy_|383-495
1slgB|W.C.	1ptsA|W.C.	1sleB|W.C.	1srfA|W.C.	1strB|W.C.	1cxe_|383-495
1stsB|W.C.	1sldB|W.C.	1srhA|W.C.	1stp_|W.C.	1cyhA|W.C.	1cgw_|383-495
1mikA|W.C.	2rmaA|W.C.	1cwaA|W.C.	1cwcA|W.C.	2rmbA|W.C.	1cgt_|383-494
1cwbA|W.C.	3cysA|W.C.	2sim_|W.C.	1gog_|151-537	1goh_|151-537	1cgv_|383-495
136 α/β domains					
1cgt_|1-382	1cxe_|1-382	1cxh_|1-382	1cxf_|1-382	1cgv_|1-382	1racA|1-150
1cgw_|1-382	1cgy_|1-382	1cgx_|1-382	1cgu_|1-382	1btb|W.C.	1rahA|1-150
1brsD|W.C.	1bgsE|W.C.	1fnd_|155-314	1fnc_|155-314	1frn_|155-314	1wsyB|W.C.
4ts1A|1-217	1tyc_|1-217	1tydE|1-217	1tybE|1-217	1tyaE|1-217	1drk_|W.C.
1cdoA|176-324	1hldA|175-324	2oxiA|175-324	1adbA|175-324	1adg_|175-324	1ctu_|1-150
1adf_|175-324	8adh_|175-324	1adcA|175-324	6adhA|175-324	1horA|W.C.	1radA|1-150

1hotA|W.C.	2secE|W.C.	1sca_|W.C.	1scnE|W.C.	1scd_|W.C.	8atcA|1-149
1scb_|W.C.	1sbc_|W.C.	1selA|W.C.	1cia_|W.C.	1pnt_|W.C.	1orb_|1-149
1bvh_|W.C.	2hnp_|W.C.	1trx_|W.C.	2tir_|W.C.	1tho_|W.C.	1dbp_|W.C.
1tkbA|535-680	1tkcA|535-680	1tkaA|535-680	1lam_|1-159	1lanA|1-159	1raeA|1-150
1bllE|1-159	1lap_|1-159	1bpm_|1-159	1bpn_|1-159	1gdtA|1-140	1acmA|1-150
3hsc_|3-188	1ngj_|4-188	1ngi_|4-188	1ngb_|4-188	1ngf_|3-188	1rhd_|1-149
1nga_|4-188	1nge_|4-188	1ngc_|4-188	1ngg_|3-188	1ngh_|4-188	1drj_|W.C.
1atr_|2-188	1ngd_|4-188	1ats_|2-188	1cde_|W.C.	1grcA|W.C.	1rafA|1-150
1cddA|W.C.	1mhtA|W.C.	1ama_|W.C.	1mag_|W.C.	1tarA|W.C.	1ttqB|W.C.
1map_|W.C.	1tasA|W.C.	1tatA|W.C.	1akaA|W.C.	1akbA|W.C.	5acn_|1-528
1akcA|W.C.	1ula_|W.C.	1amn_|W.C.	1acj_|W.C.	1acl_|W.C.	1olcA|W.C.
1ace_|W.C.	2ctc_|W.C.	5cpa_|W.C.	1cbx_|W.C.	1cps_|W.C.	1ragA|1-150
1dr1_|W.C.	1dr3_|W.C.	1dr2_|W.C.	1dr6_|W.C.	1dr4_|W.C.	1ttpB|W.C.
1dr5_|W.C.	1dr7_|W.C.	2anhA|W.C.	1hqaA|W.C.	1alkA|W.C.	2bgt_|W.C.
1ajaA|W.C.	1ajdA|W.C.	1anjA|W.C.	1aljA|W.C.	1aniA|W.C.	1olaA|W.C.
1alhA|W.C.	1ajbA|W.C.	1ajcA|W.C.	1xab_|W.C.	1ipd_|W.C.	1rabA|1-150
1hex_|W.C.	1idm_|W.C.	1raiA|1-150	1raaA|1-150		

129 α+β domains

1fut_|W.C.	2baa_|W.C.	1aec_|W.C.	2rat_|W.C.	1rpg_|W.C.	1xrc_|1-101
1rhb_|W.C.	1rnc_|W.C.	2rns_|W.C.	1rnd_|W.C.	3rn3_|W.C.	1atnD|W.C.
1rnu_|W.C.	1ras_|W.C.	1rnv_|W.C.	1rnnE|W.C.	9rsaA|W.C.	1lttA|W.C.
1rno_|W.C.	1rar_|W.C.	1rbw_|W.C.	1rnmE|W.C.	1rha_|W.C.	1xra_|1-101
1rbn_|W.C.	1sscA|W.C.	1ssbA|W.C.	1srnA|W.C.	1rpf_|W.C.	4mdhA|155-333
1rph_|W.C.	1ssaA|W.C.	1rcnE|W.C.	1rtaE|W.C.	1rtb_|W.C.	1ltgA|W.C.
1rbjA|W.C.	1rbbA|W.C.	2aas_|W.C.	1rbd_|W.C.	1rbi_|W.C.	1glv_|123-316
2rlnE|W.C.	1rbh_|W.C.	1rbe_|W.C.	1rbg_|W.C.	1rbf_|W.C.	1mrk_|W.C.
1rbe_|W.C.	1kraA|W.C.	1krbA|W.C.	1krcA|W.C.	1pgx_|W.C.	1ltbA|W.C.
1pgb_|W.C.	1pga_|W.C.	1igcA|W.C.	1fccC|W.C.	1gbl_|W.C.	3dni_|W.C.
2igg_|W.C.	2igh_|W.C.	1coy_|319-450	3monA|W.C.	1frtA|1-178	1tcs_|W.C.
1fkj_|W.C.	1fkb_|W.C.	1fkf_|W.C.	1fkl_|W.C.	2fke_|W.C.	1htlA|W.C.
1fkh_|W.C.	1fkg_|W.C.	1fkk_|W.C.	1fkiA|W.C.	1fkr_|W.C.	1dnkA|W.C.
1fks_|W.C.	1fkt_|W.C.	2secI|W.C.	1egpA|W.C.	1meeI|W.C.	1ltaA|W.C.
2tecI|W.C.	1acbI|W.C.	1egl_|W.C.	1sbnI|W.C.	1sibI|W.C.	3tms_|W.C.
3mdsA|93-203	1vig_|W.C.	1mns_|3-132	1mdr_|3-132	1grl_|137-190	1tbpA|61-155
1rldS|W.C.	1rlcS|W.C.	1comA|W.C.	2chtA|W.C.	1gaeO|149-312	1tsw_|W.C.
1mstA|W.C.	1bmsA|W.C.	1msc_|W.C.	1grb_|364-478	1gra_|364-478	3b5c_|W.C.
1gre_|364-478	1grf_|364-478	1grg_|364-478	4grl_|364-478	1lklA|W.C.	1tsy_|W.C.
1lcjA|W.C.	1lckA|117-226	1sphA|W.C.	2hpr_|W.C.	1sceA|W.C.	1tys_|W.C.
1setA|111-421	1sesA|111-421	1serA|111-421	2tscA|W.C.	1tsdA|W.C.	1tsv_|W.C.
2bbqA|W.C.	1synA|W.C.	1tsx_|W.C.			

Table A.10 The 1189 Protein Domains.

222 all-α domains

1aab_|W.C.	1cnt1|W.C.	1gh1A|W.C.	1lis_|W.C.	1prcC|W.C.	1zymA|22-144
1ab3_|W.C.	1coo_|W.C.	1gks_|W.C.	1lki_|W.C.	1pueE|W.C.	256BA|W.C.
1abv_|W.C.	1copD|W.C.	1glm_|W.C.	1lla_|110-379	1r69_|W.C.	2abk_|W.C.
1aca_|W.C.	1cpcA|W.C.	1gln_|306-468	1lla_|2-109	1rcd_|W.C.	2bct_|W.C.
1acp_|W.C.	1cpcB|W.C.	1glqA|79-209	1lliA|W.C.	1rec_|W.C.	2bmhA|W.C.
1adr_|W.C.	1cpq_|W.C.	1gnwA|86-211	1lpe_|W.C.	1res_|W.C.	2ccyA|W.C.
1adt_|176-265	1cpt_|W.C.	1grj_|2-79	1lre_|W.C.	1rfbA|W.C.	2cyp_|W.C.
1aep_|W.C.	1crkA|1-98	1grl_|410-523	1lrv_|W.C.	1rgb_|W.C.	2end_|W.C.
1af8_|W.C.	1csgA|W.C.	1grl_|6-136	1mbd_|W.C.	1ribA|W.C.	2gstA|85-217
1afrA|W.C.	1csh_|W.C.	1hbg_|W.C.	1mdyA|W.C.	1rlr_|10-221	2hmqA|W.C.
1agrE|W.C.	1csmA|W.C.	1hc2_|136-398	1mhlA|W.C.	1rom_|W.C.	2hmx_|W.C.
1aj3_|W.C.	1cuk_|156-203	1hc2_|5-135	1mhlC|W.C.	1rpo_|W.C.	2hts_|W.C.
1ak4C|W.C.	1cuk_|65-142	1hcrA|W.C.	1mmoB|W.C.	1rro_|W.C.	2int_|W.C.
1allA|W.C.	1cyi_|W.C.	1hdj_|W.C.	1mmoD|W.C.	1ryt_|2-147	2lefA|W.C.
1an2A|W.C.	1djxA|200-298	1hmcA|W.C.	1mmoG|W.C.	1scmB|W.C.	2lhb_|W.C.
1aofA|36-133	1dnpA|201-469	1hme_|W.C.	1mngA|1-92	1setA|1-110	2ligA|W.C.
1aorA|211-605	1dprA|3-64	1hnr_|W.C.	1mntA|W.C.	1sfe_|93-176	2mtaC|W.C.
1aoy_|W.C.	1dprA|65-136	1hrzA|W.C.	1mykA|W.C.	1sig_|W.C.	2mysB|W.C.
1aru_|W.C.	1dvh_|W.C.	1hstA|W.C.	1ner_|W.C.	1sly_|1-450	2pde_|W.C.
1bbhA|W.C.	1eca_|W.C.	1hueA|W.C.	1ngr_|W.C.	1sra_|W.C.	2pgd_|177-473
1bbl_|W.C.	1eciA|W.C.	1hulA|W.C.	1nkl_|W.C.	1tadA|57-177	2sas_|W.C.
1bcfA|W.C.	1ecmA|W.C.	1huw_|W.C.	1occE|W.C.	1tafA|W.C.	2sblB|150-839
1beo_|W.C.	1enh_|W.C.	1hvd_|W.C.	1occH|W.C.	1tafB|W.C.	2scpA|W.C.
1bfmA|W.C.	1erc_|W.C.	1hyp_|W.C.	1octC|5-75	1tcoB|W.C.	2spcA|W.C.
1bgc_|W.C.	1erd_|W.C.	1ihfB|W.C.	1olgA|W.C.	1tf4A|1-460	2tct_|2-67
1bia_|1-63	1erp_|W.C.	1ilk_|W.C.	1opc_|W.C.	1tfr_|183-305	2wrpR|W.C.
1bip_|W.C.	1ery_|W.C.	1imq_|W.C.	1osa_|W.C.	1tns_|W.C.	351c_|W.C.
1bmfA|380-510	1etpA|1-92	1ithA|W.C.	1oxa_|W.C.	1tpt_|1-70	3inkC|W.C.
1bmfD|358-475	1etpA|93-190	1jkw_|11-161	1pbwA|W.C.	1utg_|W.C.	3sdhA|W.C.
1bucA|233-383	1fapB|W.C.	1jkw_|162-287	1pdnC|W.C.	1vii_|W.C.	4icb_|W.C.
1bvp1|1-120	1fdcD|1-80	1jli_|W.C.	1phb_|W.C.	1vnc_|W.C.	5eas_|221-548
1bvp1|255-349	1fcdD|81-174	1jvr_|W.C.	1pnbA|W.C.	1vtmP|W.C.	5eas_|24-220
1c5a_|W.C.	1fipA|W.C.	1lbd_|W.C.	1pnbB|W.C.	1xgsA|195-271	1ash_|W.C.
1cc5_|W.C.	1fjlA|W.C.	1lbu_|1-83	1pnrA|3-58	1xsm_|W.C.	1ytfD|5-54
1cem_|W.C.	1flp_|W.C.	1lccA|W.C.	1poa_|W.C.	1yrnA|W.C.	1pprM|157-312
1cpgA|138-205	1fow_|W.C.	1lea_|W.C.	1poc_|W.C.	1yrnB|W.C.	1lh1_|W.C.
1clc_|135-575	1fps_|W.C.	1lfb_|W.C.	1pprM|1-156	1ytfB|W.C.	1gab_|W.C.
1cmbA|W.C.					

294 all-β domains

1abrB|1-140	1clc_|35-134	1gtrA|339-547	1nbcA|W.C.	1smpI|W.C.	2bb2_|86-175
1abrB|141-267	1cpn_|W.C.	1gzi_|W.C.	1nciA|W.C.	1sriA|W.C.	2bbkH|W.C.
1agjA|W.C.	1cskA|W.C.	1havA|W.C.	1neu_|W.C.	1sro_|W.C.	2bbvA|W.C.
1ah9_|W.C.	1ctm_|1-167	1hbp_|W.C.	1nfa_|W.C.	1sso_|W.C.	2bpa1|W.C.
1ahsA|W.C.	1ctm_|168-230	1hc2_|399-653	1noa_|W.C.	1stmA|W.C.	2bpa2|W.C.
1aizA|W.C.	1ctm_|231-250	1hcd_|W.C.	1npoA|W.C.	1sty_|W.C.	2cas_|W.C.
1aly_|W.C.	1ctn_|24-132	1hgeA|W.C.	1nscA|W.C.	1sva1|W.C.	2cbp_
1amy|347-403	1cto_|W.C.	1hms_|W.C.	1obpA|W.C.	1svb_|303-395	2cnd_|11-124
1anu_|W.C.	1cuk_|1-64	1hoe_|W.C.	1occB|91-227	1tdtA|W.C.	2cpl_|W.C.

1aofA|134-567	1cur_|W.C.	1hsq_|W.C.	1ospO|W.C.	1ten_|W.C.	2eng_|W.C.
1aol_|W.C.	1cwpA|W.C.	1htp_|W.C.	1pcl_|W.C.	1tf4A|461-605	2fgf_|W.C.
1aonO|W.C.	1cyx_|W.C.	1hxn_|W.C.	1pdr_|W.C.	1thjA|W.C.	2hft_|107-211
1aozA|1-129	1dar_|283-400	1i1b_|W.C.	1pex_|W.C.	1thw_|W.C.	2hft_|1-106
1aozA|130-338	1ddt_|381-535	1idaA|W.C.	1pfsA|W.C.	1tie_|W.C.	2ila_|W.C.
1aozA|339-552	1dkgA|139-197	1idk_|W.C.	1pgs_|141-314	1tiiD|W.C.	2kauB|W.C.
1arb_|W.C.	1dlc_|290-499	1ifc_|W.C.	1pgs_|4-140	1tiu_|W.C.	2kauC|2-129
1asyA|68-204	1dupA|W.C.	1ihwA|W.C.	1pht_|W.C.	1tlk_|W.C.	2kauC|423-475
1bbpA|W.C.	1dutA|W.C.	1ilr1_|W.C.	1pkyA|70-167	1tme1|W.C.	2mev1|W.C.
1bbt1|W.C.	1dynA|W.C.	1irsA|W.C.	1plc_|W.C.	1tnfA|W.C.	2mev2|W.C.
1bbt3|W.C.	1eagA|W.C.	1iyu_|W.C.	1pls_|W.C.	1tnm_|W.C.	2ncm_|W.C.
1bdo_|W.C.	1eal_|W.C.	1jdc_|358-418	1pmi_|W.C.	1tnrA|W.C.	2ohxA|1-163
1bebA|W.C.	1ebpA|10-116	1jer_|W.C.	1pms_|W.C.	1tsp_|W.C.	2ohxA|340-374
1bglA|220-333	1eft_|213-312	1kapP|247-470	1ppi_|404-496	1tul_|W.C.	2pcdA|W.C.
1bglA|3-219	1eft_|313-405	1kcw_|1-192	1prr_|1-90	1tupA|W.C.	2pcdM|W.C.
1bglA|626-730	1epbA|W.C.	1kcw_|193-338	1prr_|91-173	1ulo_|W.C.	2pec_|W.C.
1bglA|731-1023	1epnE|W.C.	1kcw_|347-553	1prtD|W.C.	1vcaA|1-90	2phlA|11-210
1bhgA|22-225	1esfA|1-120	1kcw_|554-705	1prtF|W.C.	1vcaA|91-199	2phlA|220-381
1bhgA|226-328	1eta1|W.C.	1kcw_|706-884	1pse_|W.C.	1vfbA|W.C.	2pia_|1-103
1bia_|71-317	1eur_|W.C.	1kcw_|892-1040	1pvc1|W.C.	1vie_|W.C.	2prd_|W.C.
1bmfA|24-94	1exg_|W.C.	1kevA|1-139	1pvc2|W.C.	1vmoA|W.C.	2rspA|W.C.
1bmfD|9-81	1fdr_|2-100	1kevA|314-351	1pvc3|W.C.	1wapA|W.C.	2sblB|7-149
1bncA|331-446	1fgp_|W.C.	1kit_|217-346	1pyp_|W.C.	1wba_|W.C.	2sil_|W.C.
1bovA|W.C.	1fivA|W.C.	1kit_|25-216	1qba_|28-200	1whi_|W.C.	2snv_|W.C.
1btkA|W.C.	1fmb_|W.C.	1kit_|347-543	1qorA|2-112	1who_|W.C.	2stv_|W.C.
1btn_|W.C.	1fna_|W.C.	1knb_|W.C.	1qorA|292-327	1wiu_|W.C.	2tbvA|W.C.
1bty_|W.C.	1fnb|19-154	1ksr_|W.C.	1rgs_|113-244	1wkt_|W.C.	2trcB|W.C.
1bvp1|121-254	1fuiA|356-591	1lac_|W.C.	1rip_|W.C.	1xnb_|W.C.	2tssA|1-93
1bw3_|W.C.	1fyc_|W.C.	1lcl_|W.C.	1rsy_|W.C.	1xsoA|W.C.	3cd4_|1-97
1cd1a|186-279	1gen_|W.C.	1lla_|380-628	1sacA|W.C.	1yaiA|W.C.	3cd4_|98-178
1cdcB|W.C.	1ggtA|516-627	1ltsD|W.C.	1scs_|W.C.	1yhb_|W.C.	3dpa_|1-124
1cdg_|407-495	1ggtA|628-729	1lxa_|W.C.	1se4_|1-121	1ytfC|W.C.	3dpa_|125-218
1cdg_|496-581	1ggtA|8-190	1lylA|14-153	1semA|W.C.	1ytfD|55-119	3hhrB|32-130
1cdg_|582-686	1ghk_|W.C.	1mai_|W.C.	1sftA|2-11	1zncA|W.C.	3nn9_|W.C.
1cgpA|9-137	1glaF|W.C.	1mjc_|W.C.	1sftA|245-383	1zxq_|1-86	3ullA|W.C.
1cid_|106-177	1gof_|1-150	1mmd_|34-79	1sgc_|W.C.	1zxq_|87-192	4aahA|W.C.
1cid_|1-105	1gof_|151-537	1mpp_|W.C.	1shcA|W.C.	2aaa_|382-476	4bcl_|W.C.
1ciy_|256-461	1gof_|538-639	1msaA|W.C.	1shg_|W.C.	2alp_|W.C.	4gcr_|1-85
1ckaA|W.C.	1gpc_|W.C.	1mspA|W.C.	1slaA|W.C.	2arcA|W.C.	4gcr_|86-174
1ckmA|239-327	1gpr_|W.C.	1mup_|W.C.	1sluA|W.C.	2aviA|W.C.	4kbpA|9-120

334 α/β domains

1aba_|W.C.	1dpgA|413-426	1gtmA|3-180	1nfp_|W.C.	1qrdA|W.C.	2at2A|1-144
1ad3A|W.C.	1dppA|W.C.	1gtrA|8-338	1nhp_|1-119	1raaA|1-150	2at2A|145-295
1add_|W.C.	1draA|W.C.	1gym_|W.C.	1nhp_|120-242	1raaA|151-310	2bgu_|W.C.
1adeA|W.C.	1dsbA|W.C.	1hdcA|W.C.	1nhp_|243-321	1rcf_|W.C.	2chr_|127-370
1adjA|326-421	1dts_|W.C.	1hgxA|W.C.	1nipA|W.C.	1reqA|2-560	2cmd_|1-145
1ag8A|W.C.	1dubA|W.C.	1hjrA|W.C.	1noyA|W.C.	1reqB|20-475	2cnd_|125-270
1ak5_|2-101	1dxy_|101-299	1hlpA|21-162	1nsj_|W.C.	1rlaA|W.C.	2ctb_|W.C.
1ak5_|222-483	1dxy_|1-100	1hmpA|W.C.	1nsyA|W.C.	1rlr_|222-748	2dkb_|W.C.
1amp_|W.C.	1e2b_|W.C.	1hmy_|W.C.	1ntr_|W.C.	1rnl_|5-142	2dln_|1-96
1amy_|1-346	1eaf_|W.C.	1hplA|1-336	1nulA|W.C.	1rpa_|W.C.	2dri_|W.C.
1art_|W.C.	1ebhA|142-436	1hpm_|189-381	1nzyA|W.C.	1rvaA|W.C.	2ebn_|W.C.

1asu_|W.C.	1eceA|W.C.	1hpm_|4-188	1obr_|W.C.	1rvvA|W.C.	2fx2_|W.C.
1atiA|395-505	1ecpA|W.C.	1hrdA|1-194	1ofgA|1-160	1sbp_|W.C.	2glt_|1-122
1ayl_|1-227	1ede_|W.C.	1hrdA|195-449	1ofgA|323-381	1scuA|1-121	2gstA|1-84
1ayl_|228-540	1edg_|W.C.	1hurA|W.C.	1opr_|W.C.	1scuA|122-288	2hnp_|W.C.
1bam_|W.C.	1edt_|W.C.	1hvq_|W.C.	1orb_|1-149	1scuB|239-388	2kauC|130-422
1bglA|334-625	1eft_|1-212	1hyhA|21- 166	1orb_|150-293	1sfe_|12-92	2kauC|476-567
1bksA|W.C.	1ego_|W.C.	1iceA|W.C.	1ordA|108-569	1sftA|12-244	2lbp_|W.C.
1bksB|W.C.	1eny_|W.C.	1iceB|W.C.	1ordA|1-107	1srrA|W.C.	2masA|W.C.
1ble_ |W.C.	1eriA|W.C.	1idm_|W.C.	1ortA|1-150	1tadA|27-56	2nacA|1-147
1bmfA|95-379	1esc_|W.C.	1ido_|W.C.	1ortA|151-335	1tahB|W.C.	2nacA|148-335
1bmfD|82-357	1fcdA|1-114	1igs_|W.C.	1oya_|W.C.	1tca_|W.C.	2nacA|336-374
1bmfG	1fcdA|115-255	1itg_|W.C.	1pauA|W.C.	1tde_|1-118	2ohxA|164-339
1bncA|1-114	1fcdA|256-327	1jdc_|1-357	1pauB|W.C.	1tde_|119-244	2olbA|W.C.
1broA|W.C.	1fdr_|101-248	1kevA|140-313	1pbe_|1-173	1tde_|245-316	2pgd_|1-176
1brsD|W.C.	1fds_|W.C.	1kfd_|324-518	1pbe_|276-391	1tfr_|12-180	2pia_|104-223
1byb_|W.C.	1fmcA|W.C.	1kifA|1-194	1pbn_|W.C.	1thtA|W.C.	2reb_|3-268
1cb2A|W.C.	1fnb_|155-314	1kifA|288-339	1pbp_|W.C.	1tib_|W.C.	2rn2_|W.C.
1cbg-|W.C.	1fua_|W.C.	1kte_|W.C.	1pda_|3-219	1tlfA|W.C.	2rslA|W.C.
1cdg_|1-406	1fuiA|1-355	1lam_|1-159	1pdo_|W.C.	1tml_|W.C.	2tmdA|1-340
1cec_|W.C.	1gal_|3-324	1lam_|160-484	1pea_|W.C.	1tpfA|W.C.	2tmdA|341-489
1cfr_|W.C.	1gal_|521-583	1lct_|W.C.	1pfkA|W.C.	1tplA|W.C.	2tmdA|490-645
1chd_|W.C.	1garA|W.C.	1ldb_|15-162	1php_|W.C.	1tpt_|71-335	2tmdA|646-729
1chmA|2-156	1gca_|W.C.	1ldg_|18-163	1phr_|W.C.	1trkA|3-337	2tprA|1-168
1coy_|4-318	1gd1O|313-333	1ldm_|1-160	1pii_|1-254	1trkA|338-534	2tprA|169-285
1cseE|W.C.	1gdhA|101-291	1lehA|1-134	1pii_|255-452	1trkA|535-680	2tprA|286-357
1ctn_|133-443	1gdhA|2-100	1lehA|135- 364	1pkyA|168-344	1udg_|W.C.	2trxA|W.C.
1ctt_|1-150	1gesA|147-262	1lfaA|W.C.	1pkyA|1-69	1v39_|W.C.	2ts1_|W.C.
1ctt_|151-294	1gesA|263-335	1lldA|7-149	1pkyA|351- 470	1vhrA|W.C.	2xis_|W.C.
1cus_|W.C.	1gesA|3-146	1lst_|W.C.	1pnrA|59-340	1vid_|W.C.	3chy_|W.C.
1cydA|W.C.	1gggA|W.C.	1lucA|W.C.	1pot_|W.C.	1vtk_|W.C.	3cla_|W.C.
1dapA|1-118	1ghr_|W.C.	1lucB|W.C.	1poxA|183-365	1whtA|W.C.	3dfr_|W.C.
1dapA|269-320	1glaG|254-499	1lvl_|1-150	1poxA|9-182	1whtB|W.C.	3pgm_|W.C.
1dar_|1-282	1glaG|4-253	1lvl_|151-265	1ppi_|1-403	1xel_|W.C.	3pmgA|1-190
1dctA|W.C.	1gln_|1-305	1lvl_|266-335	1psdA|108-295	1xvaA|W.C.	3pmgA|191-303
1deaA|W.C.	1glqA|1-78	1mek_|W.C.	1psdA|296-326	1xyzA|W.C.	3pmgA|304-420
1dhpA|W.C.	1gnd_|1-291	1mioA|W.C.	1psdA|7-107	1yasA|W.C.	3rubL|148-467
1dhr_|W.C.	1gnd_|389-430	1mioB|W.C.	1pta_|W.C.	1ybvA|W.C.	3tgl_|W.C.
1dih_|2-130	1gnwA|2-85	1mla_|198-307	1pud_|W.C.	1yptA|W.C.	5nul_|W.C.
1dih_|241-273	1gpb_|W.C.	1mla_|3-127	1pvdA|182-360	1yveI|83-307	5p21_|W.C.
1dik_|377-505	1gph1|235-465	1mmd_|2-33	1pvdA|2-181	1zymA|145-249	5rubA|138-457
1dik_|510-874	1gpmA|208-404	1mmd_|80-759	1pvuA|W.C.	1zymA|3-21	7icd_|W.C.
1dnpA|1-200	1gpmA|3-207	1mpb_|W.C.	1pxtA|28-293	2aaa_|1-381	8abp_ |W.C.
1dorA|W.C.	1grl_|191-366	1nal1|W.C.	1qapA|130-296	2acr_|W.C.	8dfr_|W.C.
1dosA|W.C.	1gseA|2-80	1nar-|W.C.	1qba_|338-780	2admA|W.C.	
1dpgA|1-181	1gtmA|181-419	1nbaA|W.C.	1qorA|113-291	2anhA|W.C.	

241 α+β domains

119l_|W.C.	1ctn_|444-516	1gpmA|405-525	1mli_|W.C.	1qbeA|W.C.	1znbA|W.C.
193l_|W.C.	1cyo_|W.C.	1grj_|80-158	1mngA|93-203	1raaB|1-100	2aak_|W.C.
1ab8A|W.C.	1dapA|119-268	1grl_|137-190	1molA|W.C.	1regX|W.C.	2act_|W.C.
1abrA|W.C.	1dar_|476-599	1grl_|367-409	1mrj_|W.C.	1ris_|W.C.	2baa_|W.C.
1acf_|W.C.	1dar_|600-689	1gtpA|W.C.	1msk_|W.C.	1sceA|W.C.	2bopA|W.C.
1adjA|2-325	1dcoA|W.C.	1gtqA|W.C.	1mut_|W.C.	1scuB|1-238	2chr_|1-126

1af5_|W.C.	1ddt_|1-187	1guaB|W.C.	1mxa_|108-231	1se4_|122-239	2chsA|W.C.
1afi_|W.C.	1def_|W.C.	1han_|133-289	1mxa_|1-102	1seiA|W.C.	2cmd_|146-312
1ag2_|W.C.	1dhmA|W.C.	1han_|2-132	1mxa_|232-383	1setA|111-421	2dln_|97-306
1ah6_|W.C.	1dih_131-240	1hfc_|W.C.	1napA|W.C.	1shaA|W.C.	2dnjA|W.C.
1ahq_|W.C.	1dik_|2-376	1hqi_|W.C.	1nhp_|322-447	1sly_|451-618	2glt_|123-316
1aihA|W.C.	1div_|1-55	1httA|4-325	1nox_|W.C.	1smnA|W.C.	2kauA|W.C.
1ak7_|W.C.	1div_|56-149	1humA|W.C.	1npk_|W.C.	1spbP|W.C.	2mnr_|3-132
1ako_|W.C.	1dlhA|3-81	1hxpA|178-348	1o7bT|W.C.	1srsA|W.C.	2ms2A|W.C.
1aop_|149-345	1dmaA|W.C.	1hxpA|2-177	1ofgA|161-322	1std_|W.C.	2phy_|W.C.
1aop_|346-425	1donA|W.C.	1iba_|W.C.	1ordA|570-730	1stfI|W.C.	2pia_|224-321
1aop_|81-145	1dpgA|182-412	1igd_|W.C.	1otfA|W.C.	1stu_|W.C.	2pldA|W.C.
1aorA|1-210	1dpgA|427-485	1iqzA|W.C.	1otgA|W.C.	1svr_|W.C.	2pnb_|W.C.
1apa_|W.C.	1ebhA|1-141	1kapP|1-246	1ounA|W.C.	1sxl_|W.C.	2polA|1-122
1aps_|W.C.	1efnB|W.C.	1kifA|195-287	1pba_|W.C.	1tbd_|W.C.	2polA|123-244
1apyA|W.C.	1eps_|W.C.	1kptA|W.C.	1pbe_|174-275	1tfe_|W.C.	2polA|245-366
1apyB|W.C.	1esfaA|121-233	1kuh_|W.C.	1pda_|220-307	1tif_|W.C.	2ptl_|W.C.
1ast_|W.C.	1esl_|1-118	1kvdA|W.C.	1pil_|W.C.	1tig_ |W.C.	2reb_|269-328
1atiA|1-394	1ezm_|W.C.	1kvdB|W.C.	1pkp_|4-77	1tpt_|336-440	2sicI|W.C.
1atlA	1fca_|W.C.	1lba_|W.C.	1pkp_|78-148	1uae_|W.C.	2tprA|358-482
1bia_|64-270	1fcdA|328-401	1lbu_|84-213	1plq_|1-126	1ubi-|W.C.	2tssA|94-194
1bncA|115-330	1fd2_|W.C.	1ldm_|161-329	1plq_|127-258	1udiI|W.C.	2u1a_|W.C.
1bp1_|1-217	1fjmA|W.C.	1lgr_|101-468	1pmaA|W.C.	1up1_|7-92	2vik_|W.C.
1bp1_|218-456	1fkd_|W.C.	1lgr_|1-100	1pmaB|W.C.	1up1_|99-182	3fib_|W.C.
1brnl_|W.C.	1frd_|W.C.	1lit_|W.C.	1pmd_|76-263	1urna_|W.C.	3pmgA|421-561
1bv1_|W.C.	1froA|W.C.	1lldA|150-319	1pnkA|W.C.	1vaoA|274-560	3rubL|22-147
1bvtA|W.C.	1fwp_|W.C.	1lml_|W.C.	1pnkB|W.C.	1vaoA|6-273	3rubS|W.C.
1cby_|W.C.	1fxrA|W.C.	1ltsA|W.C.	1poh_|W.C.	1vcc_|W.C.	4kbpA|121-432
1cd1A|7-185	1gbs_|W.C.	1ltsC|W.C.	1preA|2-84	1vhh_|W.C.	5rubA|2-137
1cewI|W.C.	1gcb_|W.C.	1lvl_|336-458	1prtA|W.C.	1vhiA|W.C.	7rsa_|W.C.
1chkA|W.C.	1gd1O|149-312	1lylA|161-502	1prtB|4-89	1vig_|W.C.	9rnt_|W.C.
1ckmA|11-238	1gesA|336-450	1mat_|W.C.	1ptf_|W.C.	1vjw_|W.C.	1ytbA|61-155
1coaI|W.C.	1ggtA|191-515	1mbb_|201-342	1put_|W.C.	1xgsA|1-194	1qba_|201-337
1coy_|319-450	1gmpA|W.C.	1mbb_|3-200	1pyaA|W.C.	1xgsA|272-295	1mla_|128-197
1crkA|99-380	1gnd_|292-388	1mkaA|W.C.	1qapA|8-129	1xxaA|W.C.	1gph1|1-234
1ctf_|W.C.					

Table A.11 The 25PDB Protein Domains.

443 all-α domains					
1a1w_|W.C.	1dvkB|W.C.	1h9eA|W.C.	1jr5A|W.C.	1nd9A|W.C.	1qqiA|W.C.
1a56_|W.C.	1dvoA|W.C.	1hbkA|W.C.	1jr8A|W.C.	1neq_|W.C.	1qv1A|W.C.
1a6m_|W.C.	1e29A|W.C.	1hciA|272-396	1jumA|2-72	1ng7A|W.C.	1qwnA|412-522
1ab3_|W.C.	1e52A|W.C.	1hcrA|W.C.	1jumA|73-187	1ngnA|W.C.	1qz4A|W.C.
1abv_|W.C.	1e6bA|88-220	1hd6A|W.C.	1jvr_|W.C.	1nh2B|W.C.	1r2aA|W.C.
1aduB|180-265	1e6iA|W.C.	1he8A|525-725	1jw2A|W.C.	1nhm_|W.C.	1r4aE|W.C.
1aipH|3-53	1e7lA|104-157	1hfeS|W.C.	1jybA|2-147	1ni8A|W.C.	1r4gA|W.C.
1aj3_|W.C.	1eb7A|1-164	1hh8A|W.C.	1k04A|W.C.	1nk2P|W.C.	1r5iD|W.C.
1ak0_|W.C.	1eb7A|165-323	1hkqA|W.C.	1k0mA|92-240	1nkd_|W.C.	1r5rA|W.C.
1alu_|W.C.	1eca_|W.C.	1hloA|W.C.	1k1vA|W.C.	1nkl_|W.C.	1res_|W.C.
1aoy_|W.C.	1eciA|W.C.	1hm7A|W.C.	1k3xA|125-213	1nkuA|W.C.	1rfbA|W.C.
1ash_|W.C.	1ef4A|W.C.	1hmwA|26-335	1k5oA|W.C.	1nlxA|W.C.	1rkcA|1-128
1avoA|W.C.	1elkA|W.C.	1hns_|W.C.	1k61D|W.C.	1nom_|91-148	1rkcA|129-258
1b0nA|1-68	1elrA|W.C.	1hq1A|W.C.	1k6kA|W.C.	1np7A|205-483	1rqtA|W.C.
1b0nA|74-108	1enwA|W.C.	1hqbA|W.C.	1k8kE|W.C.	1nq4A|W.C.	1rrtA|9-230
1b0nB|W.C.	1eo0A|W.C.	1hryA|W.C.	1k94A|W.C.	1ns1A|W.C.	1rsoA|W.C.
1b22A|W.C.	1eoqA|W.C.	1hs5A|W.C.	1k99A|W.C.	1nwnA|W.C.	1rsoB|W.C.
1b28A|W.C.	1erd_|W.C.	1hs7A|W.C.	1ka8A|W.C.	1ny9A|W.C.	1rss_|W.C.
1b4uA|W.C.	1eteD|W.C.	1hx8B|167-299	1kanA|126-253	1nyaA|W.C.	1rykA|W.C.
1b8zA|W.C.	1eumA|W.C.	1hx8B|22-162	1kbhA|W.C.	1nzpA|W.C.	1s0pA|W.C.
1bal_|W.C.	1exjA|3-120	1hxgA|15-220	1keyC|W.C.	1o4xA|110-163	1s7aA|W.C.
1bax_|W.C.	1eyhA|W.C.	1hxgA|221-548	1kf6B|106-243	1o4xA|5-79	1sig_|W.C.
1bbhA|W.C.	1f4iA|W.C.	1hz4A|W.C.	1kftA|W.C.	1o82A|W.C.	1sknP|W.C.
1bbn_|W.C.	1f5qB|147-252	1i1sA|W.C.	1kgzB|12-80	1o9rA|W.C.	1sly_|1-450
1bc9_|W.C.	1f5qB|6-146	1i27A|W.C.	1khoA|1-249	1oafA|W.C.	1t5jA|W.C.
1bea_|W.C.	1f6vA|W.C.	1i2tA|W.C.	1kjs_|W.C.	1oaiA|W.C.	1tafA|W.C.
1bg8A|W.C.	1f7cA|W.C.	1i4zA|W.C.	1ko9A|136-323	1oczE|W.C.	1tbaA|W.C.
1bgf_|W.C.	1fadA|W.C.	1iapA|W.C.	1koyA|W.C.	1ohzB|W.C.	1tfb_|111-207
1bh8B|W.C.	1fafA|W.C.	1ib1A|W.C.	1kqmB|W.C.	1omrA|W.C.	1ub9A|W.C.
1bh9A|W.C.	1fexA|W.C.	1ichA|W.C.	1ks8A|W.C.	1on7B|W.C.	1ucpA|W.C.
1bk6A|W.C.	1ff1A|W.C.	1ie9A|W.C.	1kwfA|W.C.	1oohA|W.C.	1ucrB|W.C.
1bkrA|W.C.	1ffkS|W.C.	1ifyA|W.C.	1kx7A|W.C.	1oqpA|W.C.	1ucvA|W.C.
1bl0A|63-124	1fipA|W.C.	1ig6A|W.C.	1l3pA|W.C.	1or6A|W.C.	1ufiB|W.C.
1bl0A|9-62	1fliA|W.C.	1iieA|W.C.	1l9lA|W.C.	1or7F|W.C.	1uk5A|W.C.
1bo9A|W.C.	1fp2A|8-108	1iioA|W.C.	1lb3A|W.C.	1orgA|W.C.	1uqvA|W.C.
1bp3A|W.C.	1fpoC|1-76	1ijyA|W.C.	1lbu_|1-83	1os6A|W.C.	1ustA|W.C.
1br0A|W.C.	1fpoC|77-171	1ik7B|W.C.	1ld8A|W.C.	1oslA|W.C.	1utg_|W.C.
1bshA|87-138	1fqkA|61-181	1irdB|W.C.	1lddA|W.C.	1otkA|W.C.	1uw4B|W.C.
1bt6A|W.C.	1fr2A|W.C.	1irg_|W.C.	1lea_|W.C.	1otrA|W.C.	1uzcA|W.C.
1bu2A|22-148	1fs9A|W.C.	1irjD|W.C.	1liaA|W.C.	1otwA|W.C.	1v38A|W.C.
1buyA|W.C.	1fyjA|W.C.	1irl_|W.C.	1lj9A|W.C.	1oyiA|W.C.	1v3f_|W.C.
1bw6A|W.C.	1fzpB|W.C.	1irqA|W.C.	1lmb3|W.C.	1oykA|W.C.	1v54H|W.C.
1c1kA|W.C.	1g03A|W.C.	1irzA|W.C.	1lq1A|W.C.	1p22B|64-136	1v74B|W.C.
1c20A|W.C.	1g1eB|W.C.	1it2A|W.C.	1lriA|W.C.	1p3bA|W.C.	1v92A|W.C.
1c53_|W.C.	1g6iA|W.C.	1itf_|W.C.	1ls1A|1-88	1p3bC|W.C.	1vf6A|W.C.
1c75A|W.C.	1g7oA|76-215	1ithA|W.C.	1lwbA|W.C.	1p3bF|W.C.	1vf6C|W.C.
1c9iA|331-357	1g8eA|W.C.	1iufA|76-141	1lycA|W.C.	1p5sA|W.C.	1vii_|W.C.
1cf7A|W.C.	1g8qA|W.C.	1iuyA|W.C.	1m12A|W.C.	1p6rA|W.C.	1vls_|W.C.

1cf7B|W.C.	1ga3A|W.C.	1iw8D|W.C.	1m15A|2-95	1p8cD|W.C.	1wjfA|W.C.
1cif_|W.C.	1gakA|W.C.	1ix9A|1-90	1m1eB|W.C.	1p94A|W.C.	1wtuA|W.C.
1cmbA|W.C.	1gc6A|88-198	1j0pA|W.C.	1m1qA|W.C.	1pc2A|W.C.	1xbl_|W.C.
1cnt4|W.C.	1gjtA|W.C.	1j0tA|W.C.	1m5nS|W.C.	1pd3A|W.C.	1xo1A|186-290
1cokA|W.C.	1gkmA|W.C.	1j2jB|W.C.	1m70A|1-92	1pfvA|389-550	1ycqA|W.C.
1coo_|W.C.	1gnc_|W.C.	1j75A|W.C.	1m70A|93-190	1pgyA|W.C.	1ytfD|5-54
1copD|W.C.	1gotG|W.C.	1j7qA|W.C.	1m8yA|W.C.	1pn5A|59-151	2a0b_|W.C.
1ctj_|W.C.	1gscA85-217	1j9iA|W.C.	1m9xC|W.C.	1pnbA|W.C.	2bby_|W.C.
1cy5A|W.C.	1gsq_|76-202	1jeiA|W.C.	1mc2A|W.C.	1pnbB|W.C.	2cpgB|W.C.
1cz2A|W.C.	1gu2B|W.C.	1jfbA|W.C.	1mdyB|W.C.	1pp7U|W.C.	2eiaA|17-147
1d2vA|W.C.	1gumA|81-220	1jfiA|W.C.	1mhzG|W.C.	1pra_|W.C.	2erl_|W.C.
1d2zB|W.C.	1guxB|W.C.	1jfiB|W.C.	1mkdA|W.C.	1psrA|W.C.	2ezi_|W.C.
1d5vA|W.C.	1gvd|W.C.	1jgcA|W.C.	1mn8D|W.C.	1psyA|W.C.	2ezl_|W.C.
1d8bA|W.C.	1gxmB|W.C.	1jgsA|W.C.	1mp1A|W.C.	1puoA|5-73	2ilk_|W.C.
1d8jA|W.C.	1gyzA|W.C.	1jhgA|W.C.	1mr8A|W.C.	1puoA|93-164	2lefA|W.C.
1d8lA|65-140	1gzsB|W.C.	1jigA|W.C.	1mwbA|W.C.	1pvhB|W.C.	2lfb_|W.C.
1dgnA|W.C.	1h0tB|W.C.	1jjrA|W.C.	1mzbA|W.C.	1pzqA|W.C.	2lisA|W.C.
1dizA|100-282	1h1jS|W.C.	1jjsA|W.C.	1n1fA|W.C.	1pzrA|W.C.	2pvbA|W.C.
1dk8A|W.C.	1h31B|W.C.	1jkuA|W.C.	1n32R|W.C.	1q02A|W.C.	2sas_|W.C.
1dnyA|W.C.	1h3lB|W.C.	1jkw_|11-161	1n3kA|W.C.	1q08A|W.C.	2tmvP|W.C.
1dp3A|W.C.	1h4jB|W.C.	1jkw_|162-287	1n62D|82-160	1q2zA|W.C.	3csmA|W.C.
1dp5B|W.C.	1h4lD|W.C.	1jl7A|W.C.	1n69B|W.C.	1q8cA|W.C.	3hdhC|204-295
1dp7P|W.C.	1h6oA|W.C.	1jli_|W.C.	1n89A|W.C.	1qatA|206-298	3htsB|W.C.
1dpuA|W.C.	1h8eI|W.C.	1jniA|W.C.	1n8vA|W.C.	1qksA|9-135	3ygsP|W.C.
1dqeA|W.C.	1h97A|W.C.	1joyA|W.C.	1n9dA|W.C.	1qntA|92-176	4ctsA|W.C.
1du6A|W.C.	1h99A|54-168	1jqjD|213-333	1nc5A|W.C.	1qpmA|W.C.	

443 all-β domains

1a1x_|W.C.	1earA|1-74	1h6xA|W.C.	1k8kC|W.C.	1npuA|W.C.	1r2mA|W.C.
1a8vA|48-118	1eazA|W.C.	1havA|W.C.	1k9cA|W.C.	1nqjA|W.C.	1r6jA|W.C.
1a9v_|W.C.	1ed7A|W.C.	1hce_|W.C.	1kawA|W.C.	1nwbA|W.C.	1r6kA|W.C.
1ag4_|W.C.	1egxA|W.C.	1hcfX|W.C.	1kd6A|W.C.	1nxmA|W.C.	1r75A|W.C.
1aiw_|W.C.	1ehkB|41-168	1hdkA|W.C.	1kdmA|W.C.	1nycA|W.C.	1rhi1|W.C.
1ajw_|W.C.	1ejfA|W.C.	1he8A|353-524	1khoA|250-370	1nz9A|W.C.	1ri9A|W.C.
1am2_|W.C.	1eo2A|W.C.	1hk6A|W.C.	1kikA|W.C.	1o1uA|W.C.	1rip_|W.C.
1aol_|W.C.	1eqrA|1-106	1hkf_|W.C.	1kj2B|W.C.	1o3sA|8-137	1rk8C|W.C.
1aonO|W.C.	1ernb_|10-116	1hlc_|W.C.	1knmA|W.C.	1o4tA|W.C.	1rkrA|W.C.
1avgI|W.C.	1ethA|337-448	1hm8A|252-459	1ko6C|W.C.	1o4yA|W.C.	1rl1A|W.C.
1ax3_|W.C.	1euwA|W.C.	1hmwA|336-599	1kq1A|W.C.	1o5lA|1-129	1rocA|W.C.
1ayoA|W.C.	1ewiA|W.C.	1hmwA|600-699	1kqrA|W.C.	1o5pA|W.C.	1rqwA|W.C.
1b34B|W.C.	1exh_|W.C.	1ht6A|348-404	1ksr_|W.C.	1o6sB|W.C.	1s2bA|W.C.
1b35A|W.C.	1exsA|W.C.	1htrp|W.C.	1kt6A|W.C.	1o7iB|W.C.	1s2eA|W.C.
1b55A|W.C.	1eysH|59-259	1hu8A|W.C.	1kum_|W.C.	1od3A|W.C.	1se1A|1-125
1b9xA|W.C.	1ezgA|W.C.	1hwhB|131-237	1kv7A|171-335	1odmA|W.C.	1sfp_|W.C.
1bak_|W.C.	1f3uB|W.C.	1hwhB|32-130	1kv7A|31- 170	1oekA|W.C.	1sg3A|1-187
1bbpA|W.C.	1f53A|W.C.	1hxrB|W.C.	1kwaA|W.C.	1ofzA|W.C.	1sg3A|195-343
1bci_|W.C.	1f6oA|W.C.	1hzeA|W.C.	1kxgA|W.C.	1ogoX|202-574	1sm4A|67-207
1bdo_|W.C.	1f86A|W.C.	1i07A|W.C.	1kxlA|W.C.	1ogoX|3-201	1sr3A|W.C.
1bdyA|W.C.	1f8eA|W.C.	1i16_|W.C.	1l1cA|W.C.	1oh1A|W.C.	1ssxA|W.C.
1bhu_|W.C.	1feuD|W.C.	1i1jA|W.C.	1l1nB|W.C.	1oh4A|W.C.	1tfhB|107-210
1bj8_|W.C.	1ffkN|W.C.	1i40A|W.C.	1l1oB|W.C.	1oioA|W.C.	1tfhB|5-106
1bpv_|W.C.	1fg9E|110-221	1i4vA|W.C.	1l2hA|W.C.	1ok0A|W.C.	1tiiD|W.C.
1bqhH|W.C.	1fg9E|13-109	1i8aA|W.C.	1lb6A|W.C.	1op4A|W.C.	1tiu_|W.C.

1br9_|W.C.	1fhoA|W.C.	1i9bA|W.C.	1lf7A|W.C.	1oqkA|W.C.	1tl2A|W.C.
1bshA|1-86	1fhrA|W.C.	1iaoA|83-178	1lixB|160-261	1ou8A|W.C.	1tme1|W.C.
1bwmA|3-116	1fi2A|W.C.	1iarB|1-96	1lktA|W.C.	1ouxA|W.C.	1ttg_|W.C.
1bymA|W.C.	1fjrA|W.C.	1iarB|97-197	1lm8V|W.C.	1oy2A|W.C.	1tul_|W.C.
1c01A|W.C.	1fl0A|W.C.	1ib5A|W.C.	1lmiA|W.C.	1p0sE|W.C.	1ub4B|W.C.
1c28A|W.C.	1flmA|W.C.	1ib8A|91-164	1lplA|W.C.	1p1mA|1-49	1ucsA|W.C.
1c4rB|W.C.	1fltY|W.C.	1ibyA|W.C.	1lugA|W.C.	1p1mA|331-404	1ud8A|391-480
1c5eA|W.C.	1fmmS|W.C.	1ic1A|1-82	1luqB|W.C.	1p35C|W.C.	1uepA|W.C.
1c5fK|W.C.	1fod1|W.C.	1ic1A|83-190	1m1fB|W.C.	1p3eA|W.C.	1uffA|W.C.
1c5lL|W.C.	1fujA|W.C.	1ifc_|W.C.	1m30A|W.C.	1p4pA|W.C.	1ufxA|W.C.
1c8cA|W.C.	1fviA|190-293	1ifrA|W.C.	1m4o|W.C.	1p9uA|W.C.	1ug1A|W.C.
1c9iA|3-330	1fyc_|W.C.	1igq|W.C.	1m5zA|W.C.	1pex_|W.C.	1ujvA|W.C.
1c9oA|W.C.	1g291|241-301	1ihw|W.C.	1m7eA|W.C.	1pfbA|W.C.	1ujxA|W.C.
1c9uB|W.C.	1g291|302-372	1iisC|5-86	1mai_|W.C.	1pfsA|W.C.	1ulp_|W.C.
1cawB|W.C.	1g2bA|W.C.	1iisC|87-171	1mdaH|W.C.	1pgs_|141-314	1umiA|W.C.
1cdb_|W.C.	1g3gA|W.C.	1ikoP|W.C.	1me6A|W.C.	1pgs_|4-140	1uscA|W.C.
1ci0A|W.C.	1g43A|W.C.	1ilfA|W.C.	1mfgA|W.C.	1ph7A|205- 328	1ut4B|W.C.
1ci5A|1-95	1g5vA|W.C.	1im3D|W.C.	1mfmA|W.C.	1ph7A|36-204	1uw7A|W.C.
1cid_|106-177	1g6eA|W.C.	1irsA|W.C.	1mgqA|W.C.	1pht_|W.C.	1uz0A|W.C.
1cid_|1-105	1g6zA|W.C.	1is3A|W.C.	1mi8A|W.C.	1pinA|6-39	1v27A|W.C.
1cpm_|W.C.	1g84A|W.C.	1iwnA|W.C.	1mjuL|108-214	1pjwA|W.C.	1vie_|W.C.
1cq3A|W.C.	1g88A|W.C.	1j0sA|W.C.	1mjuL|1-107	1pk6A|W.C.	1wbc_|W.C.
1cqyA|W.C.	1g9oA|W.C.	1j3rA|W.C.	1mnnA|W.C.	1pkhB|W.C.	1whi_|W.C.
1cr5A|26-107	1gc6A|199-297	1j7vR101-206	1muzA|W.C.	1plc_|W.C.	1wkt_|W.C.
1cto_|W.C.	1gcqC|W.C.	1j7vR|2-100	1mvfD|W.C.	1pms_|W.C.	1xntA|W.C.
1cur_|W.C.	1gglA|W.C.	1jer_|W.C.	1mvxA|W.C.	1pq7A|W.C.	1ytfD|55-119
1d1nA|W.C.	1gjxA|W.C.	1jhjA|W.C.	1my7B|W.C.	1prtD|W.C.	2arcB|W.C.
1d3bA|W.C.	1gl4B|W.C.	1jjjA|W.C.	1mzkA|W.C.	1prtF|W.C.	2bpa2|W.C.
1d7pM|W.C.	1gmiA|W.C.	1jk4A|W.C.	1n0fC|W.C.	1pse_|W.C.	2dynA|W.C.
1d8lA|1-64	1gnhA|W.C.	1jm1A|W.C.	1n32L|W.C.	1pybA|W.C.	2hntE|W.C.
1dcs_|W.C.	1gp0A|W.C.	1jo8A|W.C.	1n3jA|W.C.	1q67B|W.C.	2hrvA|W.C.
1ddmA|W.C.	1gppA|W.C.	1jopA|W.C.	1n6uA|110-212	1qauA|W.C.	2ila_|W.C.
1dg6A|W.C.	1gqhD|W.C.	1jovA|W.C.	1n6uA|1-109	1qdnA|1-85	2nlrA|W.C.
1dj7B|W.C.	1gqwB|W.C.	1jq7A|W.C.	1n8bA|W.C.	1qfoA|W.C.	2sns_|W.C.
1dqgA|W.C.	1gsgP|339-547	1jsyA|176-399	1n8kA|1-163	1qksA|136-567	2stv_|W.C.
1dqiA|W.C.	1guiA|W.C.	1jsyA|6-175	1n8kA|340-374	1qleB|108-252	2tnfA|W.C.
1dqtA|W.C.	1gv9A|W.C.	1jt8A|W.C.	1nct_|W.C.	1qouB|W.C.	3chbD|W.C.
1ds1A|W.C.	1gvmF|W.C.	1jytA|W.C.	1ne3A|W.C.	1qqp4|W.C.	3dpa_|1-124
1dxmA|W.C.	1gvp_|W.C.	1k0hA|W.C.	1nepA|W.C.	1qreA|W.C.	3dpa_|125-218
1dxwA|W.C.	1gwmA|W.C.	1k2fA|W.C.	1nglA|W.C.	1qw9A|385-501	3ezmA|W.C.
1dz1A|W.C.	1gxcA|W.C.	1k3bA|W.C.	1nh0A|W.C.	1qw9A|5-17	3mspA|W.C.
1dzkA|W.C.	1gxeA|W.C.	1k3xa|1-124	1nh2C|W.C.	1qwdA|W.C.	3ncmA|W.C.
1e0lA|W.C.	1gywB|W.C.	1k45A|W.C.	1nivA|W.C.	1qwnA|523-1044	3seb_|1-121
1e44B|W.C.	1h2cA|W.C.	1k4zA|W.C.	1nkoA|W.C.	1qwyA|W.C.	3sil_|W.C.
1e5cA|W.C.	1h2nA|W.C.	1k5cA|W.C.	1nkr_|102-200	1qxmA|149-286	3vub_|W.C.
1e5uI|1-89	1h2wA|1-430	1k5jA|W.C.	1nkr_|6-101	1qxmA|4-148	4aahA|W.C.
1e9gA|W.C.	1h3zA|W.C.	1k5nA|182-276	1nls_|W.C.	1qy1A|W.C.	4hmgA|W.C.
1e9yA|106-238	1h4aX|1-85	1k5nB|W.C.	1nnxA|W.C.	1r0uA|W.C.	4ull_|W.C.
1eajB|W.C.	1h6fbB|W.C.	1k8hA|W.C.	1nofA|31-43	1r21A|W.C.	

346 α/β domains

1aba_|W.C.	1f61A|W.C.	1i24A|W.C.	1lqtB|109-324	1oc7A|W.C.	1r18A|W.C.
1ao3A|W.C.	1f9vA|W.C.	1i2zA|W.C.	1lqtB|2-108	1od6A|W.C.	1r26A|W.C.

1ay7B|W.C.	1fezA|W.C.	1i4nA|W.C.	1lqtB|325-456	1odgA|W.C.	1r2qA|W.C.
1ayl_|1-227	1ffkC|W.C.	1i4wA|W.C.	1ls1A|89-295	1odzA|W.C.	1r5pB|W.C.
1ayl_|228-540	1ffkG|W.C.	1i69B|W.C.	1lu4A|W.C.	1oftA|W.C.	1r5xA|W.C.
1b26A|179-412	1ffkL|W.C.	1i7lA|113-214	1m0iA|W.C.	1oheA|42-198	1r5yA|W.C.
1b26A|4-178	1ffkV|W.C.	1iaqB|W.C.	1m1bB|W.C.	1ohhG|W.C.	1r6dA|W.C.
1b3oA|10-109	1fo5A|W.C.	1ibsB|167-315	1m1nA|W.C.	1ojrA|W.C.	1r6hA|W.C.
1b3oA|232-499	1fovA|W.C.	1ibsB|6-166	1m1nB|W.C.	1on4A|W.C.	1rflA|W.C.
1b4uB|W.C.	1fp2A|109-352	1iibA|W.C.	1m2dA|W.C.	1ooyA|1-242	1rfvA|W.C.
1b8gB|W.C.	1fqkA|28-60	1iiwA|W.C.	1m2eA|W.C.	1ooyA|261-481	1rhqA|W.C.
1b93A|W.C.	1fsgA|W.C.	1in1A|W.C.	1m3gA|W.C.	1orhA|W.C.	1rkuA|W.C.
1bcrA|W.C.	1fvkA|W.C.	1ioiA|W.C.	1m4lA|W.C.	1ot5A|123-460	1rpa_|W.C.
1bcrB|W.C.	1fvpA|W.C.	1itqA|W.C.	1m65A|W.C.	1ovyA|W.C.	1rrf_|W.C.
1bqcA|W.C.	1fyeA|W.C.	1iu9A|W.C.	1m6bB|311-479	1p1mA|50-330	1rtqA|W.C.
1brt_|W.C.	1fztA|W.C.	1ixh_|W.C.	1m6bB|6-165	1p33C|W.C.	1ryoA|W.C.
1bvh-|W.C.	1g291|1-240	1izyA|W.C.	1m7gD|W.C.	1p4cA|W.C.	1s4pB|W.C.
1bx4A|W.C.	1g5qA|W.C.	1j2rC|W.C.	1mavA|W.C.	1p5fA|W.C.	1sfsA|W.C.
1byi_|W.C.	1g64A|W.C.	1j5sA|W.C.	1mf7A|W.C.	1p5zB|W.C.	1shuX|W.C.
1bykA|W.C.	1g66A|W.C.	1jdnA|W.C.	1mj5A|W.C.	1p6oA|W.C.	1st9A|W.C.
1c25_|W.C.	1g7eA|W.C.	1jf8A|W.C.	1mldA|1-144	1p73C|W.C.	1sx5A|W.C.
1cen_|W.C.	1g7oA|1-75	1ji3A|W.C.	1moq_|W.C.	1p74B|102-272	1t2dA1-150
1cfzA|W.C.	1g8aA|W.C.	1jikA|W.C.	1mq0A|W.C.	1p74B|1-101	1thx_|W.C.
1cp2|W.C.	1ga6A|W.C.	1jl1A|W.C.	1muwA|W.C.	1pb7A|W.C.	1ud8A|1-390
1cqg|W.C.	1gci_|W.C.	1jlsB|W.C.	1mwjA|W.C.	1pdo_|W.C.	1uehA|W.C.
1cui_|W.C.	1gin_|W.C.	1jmkO|W.C.	1mxiA|W.C.	1pfvA|176-388	1ug6A|W.C.
1cxqA|W.C.	1gklA|W.C.	1jmvA|W.C.	1n1dA|W.C.	1pfvA|4-140	1uocA|W.C.
1d2hA|W.C.	1gllO|2- 253	1jn0A|313-333	1n25A|W.C.	1pmoC|W.C.	1ursA|W.C.
1d3vA|W.C.	1gllO|254-499	1jon_|W.C.	1n2oB|W.C.	1poiB|W.C.	1us0A|W.C.
1d4oA|W.C.	1glv_|1-122	1jq3C|W.C.	1n32B|W.C.	1pwyE|W.C.	1uslA|W.C.
1d5tA|389-431	1gn1G|W.C.	1jqjD|1-209	1n3lA|W.C.	1pyoB|W.C.	1uwcA|W.C.
1dbwB|W.C.	1gph1|235-465	1jr4A|W.C.	1n4wA|9-318	1pztA|W.C.	1uzbA|W.C.
1dciA|W.C.	1gqoV|W.C.	1jsxA|W.C.	1n55A|W.C.	1q1qA|W.C.	1v2xA|W.C.
1de5B|W.C.	1grc|W.C.	1jtvA|W.C.	1n7hB|W.C.	1q7lA|W.C.	1v7rA|W.C.
1dirA|W.C.	1gscA|1-84	1jubA|W.C.	1n7iB|W.C.	1q7lD|W.C.	1v8aA|W.C.
1dl3A|W.C.	1gsgP|8-338	1jxiA|W.C.	1n8kA|164-339	1q92A|W.C.	1vguB|W.C.
1do0A|W.C.	1gsq_|1-75	1k0mA|6-91	1n9kA|W.C.	1qc9A|W.C.	1vhwF|W.C.
1dosA|W.C.	1gumA|4-80	1k7cA|W.C.	1nbwB|W.C.	1qdlB|W.C.	1vimA|W.C.
1dqzA|W.C.	1gvfA|W.C.	1k92A|1-188	1nf9A|W.C.	1qfeA|W.C.	1xo1A|19-185
1e0jA|W.C.	1gwz_|W.C.	1kgdA|W.C.	1nh7A|1-210	1qgeE|W.C.	1yacA|W.C.
1e5kA|W.C.	1h2wA|431-710	1kgzB|81-344	1nmpA|W.C.	1qgvA|W.C.	1yub_|W.C.
1e6bA|8-87	1h6jA	1ki9B|W.C.	1nn5A|W.C.	1qhhA|W.C.	2at2A|1-144
1ecxA|W.C.	1h6vC|14-170	1kicA|W.C.	1nnfA|W.C.	1qhhB|W.C.	2at2A|145-295
1edg_|W.C.	1h6vC|171-292	1kjqB|2-112	1nnuC|W.C.	1qhhC|W.C.	2pjrB|W.C.
1eexB|W.C.	1h6vC|293-366	1kmvA|W.C.	1nofA|44-320	1qj4A|W.C.	2pth_|W.C.
1efm_|12-190	1h75A|W.C.	1kngA|W.C.	1noyA|W.C.	1qkiB|11-199	2tpsA|W.C.
1efpA|2-184	1hd2A|W.C.	1kqpA|W.C.	1np6B|W.C.	1qkiB|435-449	2tsyA|W.C.
1eiwA|W.C.	1hdoA|W.C.	1kr2F|W.C.	1np7A|1-204	1qlwB|W.C.	3cla_|W.C.
1eizA|W.C.	1hg3A|W.C.	1kte_|W.C.	1nrjB|W.C.	1qmlA|W.C.	3fua_|W.C.
1em8B|W.C.	1hjqA|W.C.	1l7aA|W.C.	1nw8A|W.C.	1qnrA|W.C.	3hdhC|12-203
1eo1A|W.C.	1hlgA|W.C.	1l8oA|W.C.	1nzjA|W.C.	1qntA|6-91	3pviA|W.C.
1eomA|W.C.	1hm8A|2-251	1lc7A|W.C.	1o08A|W.C.	1qo5K|W.C.	4eugA|W.C.
1eqa_|W.C.	1hqkA|W.C.	1lixB|262-439	1o58A|W.C.	1qopB|W.C.	6pfkA|W.C.
1es9A|W.C.	1ht6A|1-347	1lixB|57-159	1o7jA|W.C.	1qtnB|W.C.	7a3hA|W.C.
1ethA|1-336	1htwA|W.C.	1lk9A|W.C.	1o7qA|W.C.	1qtwA|W.C.	7mhtA|W.C.

1excA|W.C.	1huxA|W.C.	1lkxD|W.C.	1o8xA|W.C.	1qw9A|18-384	8abp_|W.C.
1f2tB|W.C.	1hxhA|W.C.	1ll4A|36-292	1oaa_ |W.C.	1qwnA|31-411	
1f51E|W.C.	1i0dB|W.C.	1llfA|W.C.	1oboA|W.C.	1qzmA|W.C.	

441 α+β domains

169lA|W.C.	1eqrA|421-590	1iad_|W.C.	1kn6A|W.C.	1o26A|W.C.	1r29A|W.C.
1a2n_|W.C.	1euvA|W.C.	1iajB|W.C.	1ko9A|12-135	1o2fB|W.C.	1r52B|W.C.
1a2pA|W.C.	1euvB|W.C.	1iaoA|1-82	1kotA|W.C.	1o50A|77-145	1r8hC|W.C.
1a67_|W.C.	1ev0A|W.C.	1ib8A|1-90	1kp6A|W.C.	1o7bT|W.C.	1regY|W.C.
1a9nD|W.C.	1ew4A|W.C.	1ibxA|W.C.	1kpqA|W.C.	1o7nB|W.C.	1rfa_|W.C.
1aa3_|W.C.	1exjA|121-277	1id0A|W.C.	1kptA|W.C.	1o8rA|W.C.	1rjtA|W.C.
1af5-|W.C.	1f08A|W.C.	1idpA|W.C.	1kqfB|2-245	1ocyA|W.C.	1ro2A|W.C.
1aihB|W.C.	1f0zA|W.C.	1ihrA|W.C.	1kufA|W.C.	1odhA|W.C.	1rrtA|231-360
1aipH|54-196	1f2rI|W.C.	1ijkC|W.C.	1kvdB|W.C.	1of5A|W.C.	1rwzA|1-122
1ako_|W.C.	1f32A|W.C.	1ikm_|W.C.	1kveA|W.C.	1of5B|W.C.	1rwzA|123-244
1aps_|W.C.	1f40A|W.C.	1imuA|W.C.	1kznA|W.C.	1ofhG|W.C.	1ry9A|W.C.
1apzA|W.C.	1f51A|W.C.	1iouA|W.C.	1l0oA|W.C.	1oh0A|W.C.	1ryjA|W.C.
1aq4A|W.C.	1f60B|W.C.	1ipbA|W.C.	1l1pA|W.C.	1oj5A|W.C.	1s0yD|W.C.
1aqzB|W.C.	1f7lA|W.C.	1ipgA|W.C.	1l3gA|W.C.	1ojgA|W.C.	1s0yE|W.C.
1avpA|W.C.	1f96A|W.C.	1iqsA|W.C.	1l3kA|103-181	1oo5A|W.C.	1s5fA|W.C.
1ayyB|W.C.	1f9yA|W.C.	1iqzA|W.C.	1l3kA|8-91	1opd_|W.C.	1s5uB|W.C.
1b04B|W.C.	1ffk1|1-79	1iryA|W.C.	1l4zB|W.C.	1opzA|W.C.	1s79A|W.C.
1b10A|W.C.	1ffk1|80-172	1is7K|W.C.	1l5pA|W.C.	1oqjB|W.C.	1s7jA|W.C.
1b33N|W.C.	1ffkD|W.C.	1itpA|W.C.	1l9aA|W.C.	1oqqA|W.C.	1sb6A|W.C.
1b3aA|W.C.	1ffkF|W.C.	1iu3C|W.C.	1l9yA|W.C.	1oqvA|W.C.	1scjB|W.C.
1b5eA|W.C.	1ffkP|W.C.	1iujB|W.C.	1lbu_|84-213	1oqwA|W.C.	1sf0A|W.C.
1b65A|W.C.	1ffkU|W.C.	1iv3A|W.C.	1lkkA|W.C.	1otfA|W.C.	1sgoA|W.C.
1b69A|W.C.	1fjcA|W.C.	1ivzA|W.C.	1ll4A|293-354	1otgA|W.C.	1sjwA|W.C.
1b6fA|W.C.	1fm0D|W.C.	1ix9A|91-205	1ll8A|W.C.	1owtA|W.C.	1sly_|451-618
1b87A|W.C.	1fpyA|101-468	1j0gA|W.C.	1lniA|W.C.	1p0rA|W.C.	1sp4A|W.C.
1b9lA|W.C.	1fpyA|1-100	1j27A|W.C.	1lo7A|W.C.	1p0zA|W.C.	1st4A|146-337
1bnlA|W.C.	1fu6A|W.C.	1j3gA|W.C.	1lq9A|W.C.	1p1tA|W.C.	1st4A|38-145
1bob_|W.C.	1fviA|2-189	1j4wA|104-174	1ltzA|W.C.	1p22B|2-59	1t0gA|W.C.
1bxyA|W.C.	1fw9A|W.C.	1j4wA|1-74	1ly7A|W.C.	1p32B|W.C.	1t0yA|W.C.
1by2_|W.C.	1fx4A|W.C.	1j57A|W.C.	1m0vA|W.C.	1p4lD|W.C.	1t1dA|W.C.
1bysA|W.C.	1g61A|W.C.	1j6rA|W.C.	1m15A|96-357	1p4oA|W.C.	1t2dA|151-315
1bywA|W.C.	1g71A|W.C.	1j8cA|W.C.	1m4jA|W.C.	1p65A|W.C.	1tbaB|61-155
1c05A|W.C.	1gc1G|W.C.	1jatA|W.C.	1mbxD|W.C.	1p9kA|W.C.	1tig_|W.C.
1c7kA|W.C.	1gc6A|1-87	1jatB|W.C.	1mbyA|W.C.	1pa4A|W.C.	1tiiC|W.C.
1cc8A|W.C.	1gd0A|W.C.	1jbiA|W.C.	1me4A|W.C.	1pavA|W.C.	1ub1A|W.C.
1cjkB|W.C.	1gh8A|W.C.	1jc5B|W.C.	1mg4A|W.C.	1pba_|W.C.	1ufyA|W.C.
1ckjB|W.C.	1ghhA|W.C.	1jd21|W.C.	1mg7A|14-187	1pbuA|W.C.	1unnC|W.C.
1ckv_|W.C.	1gk9A|W.C.	1jd2K|W.C.	1mg7A|188-380	1pc6B|W.C.	1uq5A|W.C.
1cqmA|W.C.	1gk9B|W.C.	1jd2L|W.C.	1mhdA|W.C.	1pcfA|W.C.	1usmA|W.C.
1cv8_|W.C.	1go1A|W.C.	1jd2M|W.C.	1mhmB|W.C.	1pil_|W.C.	1uutA|W.C.
1cxyA|W.C.	1gph1|1-234	1jfmA|W.C.	1mk0A|W.C.	1pinA|45-163	1uuzB|W.C.
1czpA|W.C.	1gpqB|W.C.	1jh6A|W.C.	1mk4A|W.C.	1pqsA|W.C.	1uw4A|W.C.
1d5tA|292-388	1gtpA|W.C.	1jhsA|W.C.	1mkbA|W.C.	1prtA|W.C.	1v2yA|W.C.
1d8iA|W.C.	1gtqA|W.C.	1jidA|W.C.	1ml8A|W.C.	1prtB|4-89	1v74A|W.C.
1d9uA|W.C.	1gw5S|W.C.	1jihA|390-509	1mldA|145-313	1pugC|W.C.	1vazA|W.C.
1dchA|W.C.	1gxuA|W.C.	1jk3A|W.C.	1mogA|W.C.	1pvmB|65-142	1vcc_|W.C.
1dcjA|W.C.	1gxyA|W.C.	1jknA|W.C.	1molA|W.C.	1pytA|W.C.	1vhiB|W.C.
1def_|W.C.	1gy7B|W.C.	1jn0A|149-312	1mszA|W.C.	1pz4A|W.C.	1vi8B|W.C.

1di2B|W.C.	1gyfA|W.C.	1jnzB|W.C.	1mw4A|W.C.	1q53A|W.C.	1vih_|W.C.
1dizA|1-99	1gyxA|W.C.	1jo0A|W.C.	1mwpA|W.C.	1q5yB|W.C.	1xxcA|W.C.
1dokA|W.C.	1h0yA|W.C.	1josA|W.C.	1mwwB|W.C.	1q8lA|W.C.	2atcB1-100
1dt4A|W.C.	1h3qA|W.C.	1jrkA|W.C.	1n13C|W.C.	1q8rA|W.C.	2bopA|W.C.
1e0gA|W.C.	1h5pA|W.C.	1jrmA|W.C.	1n32C|107-207	1qb3B|W.C.	2fdn_|W.C.
1e1hA|W.C.	1h6hA|W.C.	1jruA|W.C.	1n32C|2-106	1qddA|W.C.	2fmr_|W.C.
1e1hD|W.C.	1h6kY|W.C.	1jw3A|W.C.	1n32I|W.C.	1qdnA|86-201	2igd_|W.C.
1e44A|W.C.	1h6vC|367-495	1jyoA|W.C.	1n32J|W.C.	1qfcA|W.C.	2jdxA|W.C.
1e5uI|90-187	1h8cA|W.C.	1k0kA|W.C.	1n4wA|319-450	1qg7A|W.C.	2nef_|W.C.
1e7kA|W.C.	1hbnB|2-188	1k1gA|W.C.	1n62C|1-177	1qhkA|W.C.	2nmtA|34-218
1e7lA|1-103	1he8A|144-321	1k3eA|W.C.	1n62C|178-286	1qkfA|W.C.	2pleA|W.C.
1e87A|W.C.	1hl6D|W.C.	1k4iA|W.C.	1n62D|2-81	1qkiB|200-434	2proB|4-85
1e9yA|1-105	1hmjA|W.C.	1k5nA|1-181	1n6zA|W.C.	1qkiB|450-511	2proB|86-158
1earA|75-142	1hq6A|W.C.	1k83K|W.C.	1neiA|W.C.	1qklA|W.C.	2sak_|W.C.
1eayC|W.C.	1hqi_|W.C.	1k8bA|W.C.	1nh7A|211-284	1ql0A|W.C.	2sxl_|W.C.
1eb6A|W.C.	1hqz1|W.C.	1k8kF|W.C.	1nkiA|W.C.	1qmtA|W.C.	2tbd_|W.C.
1ecsA|W.C.	1hv2A|W.C.	1k92A|189-444	1no5A|W.C.	1qolA|W.C.	2tldI|W.C.
1ef5A|W.C.	1hywA|W.C.	1kafD|W.C.	1nr3A|W.C.	1qr5A|W.C.	2u1a_|W.C.
1eggB|W.C.	1hz6B|W.C.	1kanA|1-125	1nrjA|W.C.	1qs1A|265-461	2vil_|W.C.
1egwA|W.C.	1hztA|W.C.	1kcgC|W.C.	1nskl|W.C.	1qs1A|60-264	3gcc_|W.C.
1ektA|W.C.	1i0vA|W.C.	1kcqA|W.C.	1nvjD|W.C.	1qsoA|W.C.	3lzt_|W.C.
1el6A|W.C.	1i12A|W.C.	1kf6B|1-105	1nwwB|W.C.	1qstA|W.C.	3seb_|122-238
1emwA|W.C.	1i17A|W.C.	1kg0C|W.C.	1nwzA|W.C.	1qtoA|W.C.	3znbA|W.C.
1eqkA|W.C.	1i35A|W.C.	1kjkA|W.C.	1nxiA|W.C.	1qxyA|W.C.	
1eqrA|107-287	1i7eA|W.C.	1kjqB|113-318	1nz8A|W.C.	1qymA|W.C.	
1eqrA|288-420	1i9yA|W.C.	1kn0A|W.C.	1o0pA|W.C.	1qynA|W.C.	

VITA

Fadime Üney Yüksektepe completed the high school in Salihli Sekine Evren Anatolian High School, Manisa, in 1998. She received her high honor B. Sc. and M. Sc. degrees in Chemical Engineering from Istanbul Technical University, in 2003 and in Industrial Engineering from Koç University, in 2005, respectively. Since 2005, she is in the Ph.D. program in Industrial Engineering & Operations Management at Koç University as a teaching and research assistant. For the completion of the program, she has studied the thesis “MILP Based Hyper-Box Enclosure Approach to Multi-Class Data Classification”.