อัลกอริทึมการเรียนรู้สำหรับการทำนายผลฟีโนไทป์ของการดื้อยาสำหรับเชื้อเอชไอวีชนิดที่ 1

นางสาวอนันตพร ศรีสวัสดิ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2550

LEARNING ALGORITHMS FOR PREDICTING HIV-1 PHENOTYPIC DRUG RESISTANCE

Miss Anantaporn Srisawat

A Dissertation Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy Program in Computer Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2007

Thesis Title	LEARNING ALGORITHMS FOR PREDICTING HIV-1 PHENOTYPIC DRUG RESISTANCE
By	Miss Anantaporn Srisawat
Field of Study	Computer Engineering
Thesis Advisor	Professor Boonserm Kijsirikul, Ph.D.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirements for the Doctoral Degree

..... Dean of the Faculty of Engineering
(Associate Professor Boonsom Lerdhirunwong, Dr. Ing)

THESIS COMMITTEE

..... Chairman
(Professor Prabhas Chongstitvatana, Ph.D.)

..... Thesis Advisor
(Professor Boonserm Kijsirikul, Ph.D.)

..... External Member
(Associate Professor Wasun Chantratita, Ph.D.)

..... Member
(Assistant Professor Yachai Limpiyakorn, Ph.D.)

..... Member
(Chotirat Ratanamahatana, Ph.D.)

อนันตพร ศรีสวัสดิ์ : อัลกอริทึมการเรียนรู้สำหรับการทำนายผลฟีโนไทป์ของการดื้อยาสำหรับเชื้อเอชไอวีชนิดที่ 1. (LEARNING ALGORITHMS FOR PREDICTING HIV-1 PHENOTYPIC DRUG RESISTANCE) อ. ที่ปรึกษา : ศ. ดร. บุญเสริม กิจศิริกุล, 104 หน้า.

ข้อจำกัดของการรักษาโรคเอดส์ คือการที่เชื้อไวรัสเอชไอวีมีปฏิกิริยาต่อยาลดลง ซึ่งเรียกว่าอาการดื้อยา สาเหตุของการดื้อยานี้ เกิดจากการกลายพันธุ์ของยีนที่อยู่ในส่วนเอนไซม์รีเวิรสทรานสคริปเตสและเอนไซม์โปรเตสของเชื้อเอชไอวี ดังนั้น การทดสอบการดื้อยาจึงมีบทบาทสำคัญในการรักษาผู้ป่วยเชื้อเอชไอวี ในทางการแพทย์มีวิธีการทดสอบการดื้อยาสองวิธีคือแบบจีโนไทป์และแบบฟีโนไทป์ ข้อดีของวิธีจีโนไทป์คือให้ผลการทดสอบที่เร็วกว่าและเสียค่าใช้จ่ายน้อยกว่าวิธีฟีโนไทป์ อย่างไรก็ตาม ผลที่ได้จากการทำนายด้วยวิธีฟีโนไทป์จะให้ผลการทำนายที่เข้าใจง่ายกว่าวิธีจีโนไทป์

งานวิจัยนี้ประยุกต์ใช้อัลกอริทึมการเรียนรู้สี่แบบคือ ซัพพอร์ตเวกเตอร์แมชชีน อาร์บีเอฟเน็ตเวิร์ก เคเนียเรสท์เนเบอร์และการจำแนกประเภทด้วยวิธีแอสโซซิเอชัน เพื่อสร้างโมเดลต่างๆ สำหรับจำแนกประเภทการดื้อยาของเชื้อไวรัสเอชไอวีชนิดที่ 1 จากข้อมูลจีโนไทป์ นอกจากนี้ยังได้ศึกษาพฤติกรรมในการทำนายผลการดื้อยาของอัลกอริทึมการเรียนรู้ในแต่ละอัลกอริทึม สุดท้ายงานวิจัยนี้ได้เสนอวิธีในการสร้างตัวจำแนกประเภทประกอบแบบไดนามิกส์ด้วย

งานวิจัยนี้ได้เปรียบเทียบสมรรถนะในการทำนายผลการดื้อยาของโมเดล ที่สร้างด้วยอัลกอริทึมการเรียนรู้กับระบบการทำนายผลการดื้อยาแบบออนไลน์สองระบบคือ เอชไอวีดีบี และจีโนทูฟีโน จากผลการทดลองพบว่า อัลกอริทึมการเรียนรู้ทุกตัวให้ค่าความแม่นยำในการทำนายเฉลี่ยสูงกว่าระบบเอชไอวีดีบีและจีโนทูฟีโน สำหรับการประเมินสมรรถนะในการทำนายผลการดื้อยาของตัวจำแนกประเภทประกอบแบบไดนามิกส์ที่นำเสนอนั้น งานวิจัยนี้ได้ทำการเปรียบเทียบผลกับเทคนิคการรวมตัวจำแนกประเภทสองเทคนิค คือวิธีโหวตเสียงข้างมากและวิธีการเรียนรู้แบบนาอีฟเบย์ ผลการทดลองที่ได้พบว่าวิธีการตัวจำแนกประเภทประกอบแบบไดนามิกส์ที่นำเสนอนั้นให้ค่าความแม่นยำเฉลี่ยสูงที่สุด

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต.....
สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่ออาจารย์ที่ปรึกษา.....
ปีการศึกษา 2550

4671845321 : MAJOR COMPUTER ENGINEERING

KEY WORD: PREDICT HIV-1 DRUG RESISTANCE / LEARNING ALGORITHM / DYNAMIC CLASSIFIER COMBINATION

ANANTAPORN SRISAWAT: LEARNING ALGORITHMS FOR PREDICTING HIV-1 PHENOTYPIC DRUG RESISTANCE. THESIS ADVISOR: PROF. BOONSERM KIJSIRIKUL, Ph.D., 104 pp.

The limitation of HIV treatment is the decrease of the viral sensitivity to the drug that is called drug resistance. The cause of drug resistance is from the mutations in the reverse transcriptase and protease enzymes of HIV. Thus, resistance testing plays an important role in HIV treatment. In the medical area, there are two methods of resistance testing: genotyping and phenotyping. The advantages of genotypic testing are faster and cheaper than phenotyping. On the other hand, the results of phenotypic method are easier to interpret than genotypic testing.

This thesis applied four learning algorithms, which are the Support Vector Machine (SVM), the Radial Basis Function Network (the RBF network), k-Nearest Neighbor (*k*-NN), and Classification Based on Association (CBA), to construct the models for classifying HIV-1 drug resistance from HIV-1 genotypic data. Further, the predictive behavior of each classification model was studied. Finally, a new dynamic classifier combination method was proposed to construct the composite classifier from these single models.

The predictive performances of the learning algorithms were compared with two online drug resistance prediction systems: HIVdb and Geno2Pheno. Our experimental results demonstrated that all learning algorithms yielded the higher average accuracy than that of the online systems. To evaluate the predictive performance of the proposed dynamic classifier combination method, we compared the accuracy with two classifier combination methods which are majority voting and Naïve Bayes. The results showed that our proposed method provided the best average predictive performance.

Department: Computer Engineering Student's signature: Anantaporn Srisawat

Field of study: Computer Engineering Advisor's signature:

Academic year :2007

ACKNOWLEDGEMENTS

I would like to deeply thank my thesis advisor, Professor Dr. Boonserm Kijsirikul, who accepted me to study in the Doctor of Philosophy Program of the Computer Engineering department and advised me to research on the bioinformatics area. He also endorsed my application for the research funding from National Center for Genetic Engineering and Biotechnology (BIOTEC). During my research period, he has been providing valuable guidance and comments to my work. Moreover, he helped me proofreading my publications including this thesis document. Without his care and consideration, this thesis would likely not succeed.

I would like to thank Dr. Mikael Borden, who was my advisor when I carried out my research overseas at the School of Information Technology and Electrical Engineering, University of Queensland. He suggested me some interesting ideas which are useful for researching on the bioinformatics area.

I would like to thank my thesis committee members: Professor Dr. Prabhas Chongstitvatana, Associate Professor Dr. Wasun Chantratita, Assistant Professor Dr. Yachai Limpiyakorn, and Dr. Chotirat Ratanamahatana, who provided valuable comments during the committee meetings and helped me proofreading this thesis document. I am also thankful to Associate Professor Dr. Wasun Chantratita for providing clinical-data to evaluate our proposed algorithm in this thesis.

I would like to thank BIOTEC for the funding that supported my publication expenses. In addition, I would like to thank the Thai Government for the research grant that supported me when doing the research overseas.

My thanks go to all members of the Machine Intelligence and Knowledge Discovery (MIND) Laboratory for their helps and comments on my thesis. Many thanks to Ekawat Pasomsub, who helped me in preparing amino acid sequence of protease enzymes and reverse transcriptase enzymes of HIV-1. Moreover, he gave me a great advice for the HIV-1 background.

And finally never enough thanks to my family for their supports and encouragement throughout my study. My success would not come true without their kindness and love.

CONTENTS

Page

CHAPTER

Page

CHAPTER

Table Page

Table Page

Figure Page

CHAPTER I

INTRODUCTION

This chapter presents the motivation, objectives, scopes, benefit of the work, and research methodology of the thesis.

1.1 Motivation

Nowadays, there are many antiretroviral drugs but HIV-1 therapies are still not very successful. The limitation of treatment success is the decrease of the viral sensitivity to the drug called drug resistance. The cause of drug resistance is the mutations in the reverse transcriptase and protease enzymes of HIV-1. In addition, it has been estimated that every possible single point mutation occurs between 10^4 and 10^5 times per day in an untreated HIV-1 infected individual and that double mutants also occur commonly (Coffin, 1995). Thus resistance testing plays an important role in managing HIV infections. In a medical area, there are two methods for resistance testing: genotyping and phenotyping.

Genotypic resistance testing can be performed by scanning the viral genome for resistance-associated mutations. The final results of this method provide a prediction of susceptibility, usually classified into two or more groups (e.g. sensitive, resistant or intermediate). The phenotypic testing can be performed by measuring viral activity in the presence or absence of drug. The results of phenotypic testing are usually reported as resistance factors (real values) called fold change. The fold change refers to the fraction between 50% inhibitory drug concentration value (IC_{50}) of the patient's virus to the IC_{50} value of the standardized wild type virus ($IC_{50(patient)}$/ $IC_{50(reference)}$). However, the advantages and drawbacks of these methods are different. The advantages of genotyping are faster and cheaper than phenotyping. On the other hand, the results of phenotypic method are easier to interpret than those of genotypic testing.

At present, there are public datasets of genotype-phenotype available on the websites: Stanford HIV RT and Protease Sequence Database, and thus a learning algorithm is an appropriate way to construct the model for predicting the phenotypic results. In model construction process (or learning process), this approach uses

genotype data as inputs and it produces phenotype data as the output. One of the advantages of using the learning algorithm to construct the model instead of phenotypic testing is the prediction time. Although both methods of phenotypic testing and the learning algorithm provide the same output in the format of the fold change value, the learning algorithm takes less time than phenotypic testing in prediction. It takes a few seconds to produce a result by using the model from the learning algorithm, whereas it takes several weeks for phenotypic testing. Moreover, the model generated from a learning algorithm helps reduce the cost of phenotypic testing. However, the performance of the learning algorithm depends on the amount of phenotypic training data. The more phenotypic data, the more accuracy of the learning algorithm gains.

This thesis applies the learning algorithms to construct the models for predicting HIV-1 phenotypic drug resistance from HIV-1 genotypic data. In addition, this thesis studies the predictive behavior of each classification model. Finally, a new dynamic classifier combination method is proposed to construct the composite classifier from these single models.

1.2 Objectives

The objectives of this thesis are as follows:

1. Apply four learning algorithms, i.e. the Support Vector Machine (SVM), the Radial Basis Function Network (the RBF network), k-Nearest Neighbor (*k*-NN), and Classification based on Association (CBA) to construct the models for classifying HIV-1 drug resistance from genotypic data.

2. Study the predictive behavior of each classification model constructed by these learning algorithms.

3. Propose a new classifier combination method.

1.3 Scopes

The scopes of this thesis are as follows:

1. Construct the models to classify drug resistance into two classes: resistant and susceptible for 15 drugs separately (6 Protease Inhibitors (PIs), 6 Nucleoside Reverse Transcriptase Inhibitors (NRTIs), and 3 Non-Nucleoside Reverse Transcriptase Inhibitors (NNRTIs)).
2. Assess the predictive performance of the proposed methods by using 10-fold cross-validation technique and compare to the genotypic HIV-1 resistance interpretation system and other existing methods.
3. Develop a method for classifier combination and compare the predictive accuracy of the proposed method with other classifier combination methods such as majority voting and Naïve Bayes.

1.4 Benefit of the Work

This thesis provides a new ensemble learning method for the application of the prediction of HIV-1 phenotypic drug resistance from HIV-1 genotypic data that yields a better predictive performance than existing methods.

1.5 Research Methodology

1. Study HIV-1 structure, HIV-1 drug resistance, and HIV-1 drug resistance testing.
2. Review existing researches on the prediction of phenotypic drug resistance from HIV-1 genotypes.
3. Study fundamental theories of learning algorithms and feature subset selection techniques.
4. Collect and prepare initial datasets.
5. Set up experiments and test for single classifiers.
6. Analyze the result of single classifiers.
7. Develop a new classifier combination method.
8. Analyze the result of the ensemble learning and make conclusions.

1.6 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter II describes background of the type 1 human immunodeficiency virus (HIV-1) and reviews the HIV-1 drug resistance prediction applications. In addition, the theoretical backgrounds about the learning algorithms, feature selection techniques, and classifier combination methods used in this thesis are described. In Chapter III, we explain the processes of model construction by using single classifiers, i.e. CBA, SVM, the RBF network, and *k*-NN. Chapter IV presents the new algorithm for classifier combination.

Chapter V shows the experimental results. In the first part of this chapter, we compare the predictive performance of four learning algorithms with the online drug resistance prediction systems such as HIVdb and Geno2Pheno. In addition, the predictive behaviors of each learning algorithm are analyzed in this chapter. For the latter part of this chapter, the comparison of the predictive performance between the proposed classifier combination method and other methods is demonstrated. Then the discussion of how our proposed method enhances the predictive performance of the single classifiers is presented at the end of this chapter. Finally, the conclusion of this research is presented in Chapter VI.

CHAPTER II

BACKGROUND AND LITERATURE REVIEW

The context of this chapter is divided into two main sections. The background of the type 1 human immunodeficiency virus (HIV-1) is described in the first section. In addition, the theoretical backgrounds about the learning algorithms, feature selection techniques used in this thesis, and the background of a composite classifier are explained. For the latter section, the literature reviews of the HIV-1 drug resistance prediction applications are reported. Moreover, applications in bioinformatics area which use a classifier combination method are reviewed.

2.1 HIV-1 Background

This section explains the general descriptions of HIV-1 which are HIV-1 life cycle, antiretroviral agents, and drug resistance testing.

2.1.1 HIV-1 Life Cycle

There are six steps of HIV-1 life cycle as shown in Figure 2.1.

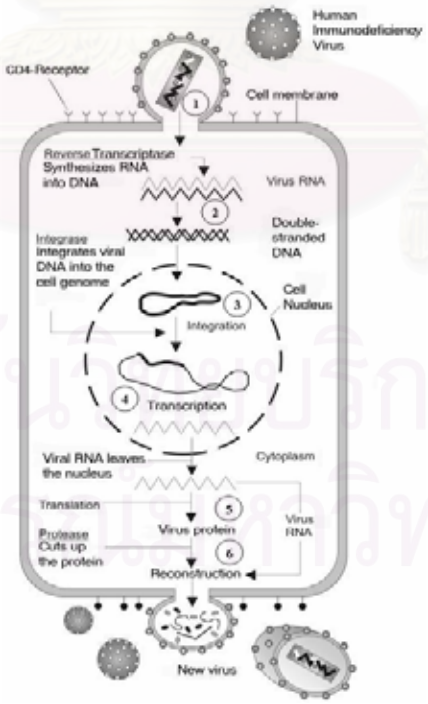


Figure 2.1: HIV-1 life cycle (Wikipedia, 2008).

2.1.1.1 Binding

HIV begins its infection of a susceptible host cell by binding to the CD4+ receptor on the host cell. When HIV binds to a CD4+ surface receptor, it activates other proteins on the cell's surface, allowing the HIV envelope to fuse to the outside of the cell. After binding process, the viral capsid which contains the RNA and important enzyme, is released into the host cell.

2.1.1.2 Reverse Transcription

At this step, HIV is stabilized by copying RNA into DNA and inserting it into the host cell's chromosomes. This means the virus can perform more subtle functions by using the host transcription machinery. The virus generates DNA from the HIV RNA using the reverse transcriptase enzyme to perform reverse transcription.

2.1.1.3 Integration

The viral DNA is carried to the host cell's nucleus. After that, the viral DNA must be integrated into the host cell DNA using the integrase enzyme. This new DNA is called proviral DNA. If the proviral DNA becomes integrated into the host cell's DNA the cell is now fully infected but not actively producing HIV proteins. This is the latent stage of an HIV infection.

2.1.1.4 Transcription

Once HIV's genetic material is inside the host cell's nucleus, it directs the cell to produce new HIV. The strands of viral DNA in the nucleus separate and special enzyme create a complementary strand of genetic material called messenger RNA or mRNA.

2.1.1.5 Translation

The mRNA carries instructions for making new viral proteins from the nucleus to a kind of workshop in the cell. Each section of the mRNA corresponds to a protein building block for making a part of HIV. As each mRNA strand is processed, a corresponding string of proteins is made. This process continues until the mRNA strand has been transformed into new viral proteins needed to make a new virus.

2.1.1.6 Viral Assembly and Maturation

The final step begins with the assembly of new virus. Long strings of proteins are cut off by protease enzyme. These proteins serve a variety of functions; some become structural of new HIV, while others become enzymes.

Once the new viral particles are assembled, they bud off the host cell, and create a new virus. The virus then enters the maturation stage, which involves the processing of viral proteins. Maturation is the final step in the process and is required for the virus to become infectious. With viral assembly and maturation complete, the virus is able to infect new cells. Each infected cell can produce a lot of new viruses.

2.1.2 Antiretroviral for HIV-1

There are three classes of antiretroviral drugs that we used in our thesis: protease inhibitors (PIs), nucleoside reverse transcriptase inhibitors (NRTIs), and non-nucleoside reverse transcriptase inhibitors (NNRTIs).

2.1.2.1 Nucleoside Reverse Transcriptase Inhibitors (NRTIs)

The first class of drugs approved by the FDA is NRTI. There are several drugs in the NRTI class such as zidovudine (AZT), didanosine (ddI), zalcitabine (ddC), stavudine (d4T), lamivudine (3TC), and abacavir (ABC). NRTIs work by binding to reverse transcriptase enzyme in the reverse transcription step (step 2). NRTIs contain faulty versions of the building blocks used by reverse transcriptase to convert RNA to DNA. When reverse transcriptase uses these faulty building blocks, the new DNA cannot be built correctly (Seattle Treatment Education Project, May, 2000).

2.1.2.2 Non-Nucleoside Reverse Transcriptase Inhibitors (NNRTIs)

There are currently three drugs approved for use in this class: nevirapine (NVP), delavirdine (DLV), and efavirenz (EFV). NNRTIs work by attaching themselves to reverse transcriptase enzyme to prevent the enzyme from converting RNA to DNA. “In turn, HIV’s genetic material cannot be incorporated into the healthy genetic material of the cell, and prevents the cell from producing new virus” (Seattle Treatment Education Project, June, 2000). However NNRTIs work in is the same point in the life cycle interfered with by NRTIs. The difference is that NNRTIs simply do it in a different way.

2.1.2.3 Protease Inhibitors (PIs)

Currently, there are eight approved protease inhibitors which are amprenavir (APV), atazanavir (ATV), indinavir (IDV), lopinavir (LPV), nelfinavir (NFV), ritonavir (RTV), saquinavir (SQV), and fosamprenavir (FPV). PIs work by blocking the activity of the protease enzyme in viral assembly step (step 6). When the PIs bind to the protease enzyme, the new viruses still leave the cell, but they are unable to infect other cells (James and Pharm, 2005).

2.1.3 Resistance Testing

“Human immunodeficiency virus or HIV is a retrovirus that causes Acquired Immune Deficiency Syndrome (AIDS), a condition in which the immune system begins to fail, leading to life-threatening opportunistic infections” (Wikipedia). HIV-1 is one species of human-infecting HIV. It is thought to have originated in southern Cameroon after jumping from wild chimpanzees to humans during the twentieth century. HIV-1 is the most virulent since it is easily transmitted and is the cause of the majority of HIV infection globally.

The objective of the antiretroviral therapy is to prevent disease progression and prolong survival, while maintaining quality of life. It is expected that long-term nonprogressive will be achieved by reducing plasma viral load as much as possible for as long as possible. The use of combinations of antiretrovirals with no overlapping

toxicity and demonstrated antiviral synergy is recommended to maximize the duration of the antiviral response (Yeni, et al., 2002).

Although there are many antiretroviral drugs, HIV-1 therapies are still not very successful. The limitation of treatment success is the decrease of the viral sensitivity to the drug called drug resistance. The cause of drug resistance is the mutations in the reverse transcriptase (RT) and protease enzymes of HIV-1. In addition, "it has been estimated that every possible single point mutation occurs between 10^4 and 10^5 times per day in an untreated HIV-1 infected individual and that double mutants also occur commonly" (Coffin, 1995). Thus resistance testing is an important role in management of HIV infections.

Currently there are two methodologies for resistance testing: genotyping and phenotyping (Demeter and Haubrich, 2001).

2.1.3.1 Genotypic Testing

For genotyping, resistance testing can be performed by scanning the viral genome for resistance-associated mutations. The results of this method are obtained by using specific software that facilitates the process of sequence alignment and summarizes codon changes. Interpretation of results from genotypic assays requires knowledge of the association of specific mutations with either phenotypic resistance or virologic response to a given drug called rules-based algorithms. These algorithms provide a prediction of susceptibility, usually classified into two or more groups (e.g. sensitive, resistant, and intermediate).

2.1.3.2 Phenotypic Testing

The phenotypic testing can be performed by measuring viral activity in the presence and absence of drug. This method measures the ability of HIV-1 to grow in the presence of different antiretroviral agents over a fixed period in cell culture. The results of phenotypic testing are usually reported as resistance factors (real value) called fold change. The fold change refers to the fraction between 50% inhibitory drug concentration value (IC_{50}) of the patient's virus to the IC_{50} value of the standardized wild

type virus ($IC_{50(patient)}$/ $IC_{50(reference)}$). If the fold change is above a certain value called cutoff, the virus is resistant to that drug.

The advantages of genotyping are faster and cheaper than phenotyping since it is less complex. But the disadvantage of genotyping is the difficulty to translate the results into a meaningful conclusion about the resistance of the virus to drugs. On the other hand, the results of phenotypic method are easier to interpret than genotypic testing because the phenotypic results are represented by a single number for each drug. However, the phenotypic method procedure is relatively complex, so it takes a longer time than the genotypic method to produce accurate results from ten days to several weeks. Moreover, the intricacy of this test also makes it more expensive.

2.2 Theoretical Backgrounds of Learning Algorithms

2.2.1 Association Rule Mining

Association rule mining is a useful technique for discovering correlation among items. This approach was first introduced for market basket analysis (Agrawal, Imielinski and Sawami, 1993).

Let $I=\{i_1,i_2,\ldots,i_3\}$ be a set of literals called items and let the database consist of a set of transactions. An association rule has the form $X \rightarrow Y$ where $X \subset I$, $Y \subset I$, and $X \cap Y = \varnothing$. X is called an antecedence and Y is a consequence of the rule. For example of a purchases relation, the rule {pen}$\rightarrow${ink} means “if a pen is purchased in a transaction, it is likely that ink will also be purchased in that transaction”.

There are two important measures used to select the interesting association rules:

- Support: The support for a set of items is the percentage of transactions containing both X and Y.
- Confidence: The confidence for the rule $X \rightarrow Y$ is the percentage of transactions containing X that also contain Y.

Considering the rule {pen}$\rightarrow${ink} again, if the support of this rule is 75 percent, and the confidence is 95 percent, it can make the observation: “in 75 percent

of the transactions both a pen and ink are purchased together, and 95 percent of the transactions that contain a pen also contain ink".

Association rule mining is the process of generating all interesting rules that satisfy the user-specified *minimum support* (*minsup*) and *minimum confidence* (*minconf*). There are two steps in association rule mining process: discovering frequent itemsets and generating association rules.

2.2.1.1 Discovering Frequent Itemsets

This process finds all sets of items that have transaction support above *minsup*. The support for an itemset is the number of transactions that contain the itemset. Itemsets satisfying *minsup* are called frequent itemsets.

For discovering all frequent itemsets, the Apriori algorithm (Agrawal and Srikant, 1994) is used to generate frequent itemsets. An important property of the Apriori is that every subset of a *frequent* itemset must also be a frequent itemset.

```
1)  L_1 = {large 1-itemsets};
2)  for ( k = 2; L_{k-1} ≠ ∅; k++ ) do begin
3)     C_k = apriori-gen(L_{k-1}); // New candidates
4)     forall transactions t ∈ D do begin
5)        C_t = subset(C_k, t); // Candidates contained in t
6)        forall candidates c ∈ C_t do
7)           c.count++;
8)     end
9)     L_k = {c ∈ C_k | c.count ≥ minsup}
10) end
11) Answer = ⋃_k L_k;
```

Figure 2.2: The Apriori algorithm (Agrawal and Srikant, 1994).

The Apriori algorithm is shown in Figure 2.2. Let L_k be a set of frequent *k*-itemsets. The word *k*-itemsets means an itemset having *k* items. Let C_k be a set of candidate *k*-itemsets and *D* be a set of transactions.

The Apriori algorithm makes multiple passes over the transactions for finding frequent itemsets. In the first pass, the algorithm counts the support of individual items and determines which of them are frequent 1-itemsets. A subsequent pass (or

pass *k*) consists of two phases. First, the frequent itemsets L_{k-1} found in the (*k*-1)th pass are used to generate the candidate itemsets C_k, using the *apriori-gen* function. The *apriori-gen* function generates the candidate itemsets having *k*-items by joining frequent itemsets having *k*-1 items, and deleting those that contain any subset which is not frequent. Next, the database is scanned and the support of candidates in C_k is counted. The *subset* function is used for fast counting of candidates in C_k. At the end of the pass, the algorithm determines which of candidate itemsets are actually frequent itemsets, and uses them as the seeds for the next pass. This process continues until no new *frequent* itemsets are found.

2.2.1.2 Generating Association Rules

Once frequent itemsets are identified, the generation of all possible rules with the user-specified *minconf* is straightforward. To generate a candidate rule from frequent itemset X, X is divided into two itemsets as a form "a $\rightarrow$ (X-a)". If the ratio of support(X) to support(a) of the candidate rule is at least *minconf*, this process will output this rule.

2.2.2 Classification Based on Associations (CBA)

Associative classification is the first integrated framework of classification rule mining and association rule mining (Liu, Hsu and Ma, 1998). The aim of this framework is to make association rule mining technique applicable to classification tasks. The integration is done by focusing on a special subset of association rules whose right-hand-side are restricted to the classification class attribute. The special subset of rules is called Class Association Rules (CARs). This framework adopts an existing association rule mining algorithm to mine all the CARs that satisfy the *minsup* and *minconf* constraints. For generating the complete set of CARs, Liu, Hsu and Ma (1998) proposed a new algorithm called Classification Based on Associations (CBA). The CBA algorithm consists of two parts: a rule generator called CBA-RG and a classifier builder called CBA-CB.

2.2.2.1 The CBA-RG Algorithm

Let <*condset*, *y*> be a form of a *ruleitem*, where *condset* is a set of items, and y∈Y is a class label. *condsupCount* is the number of cases in *D* that contain the *condset*. *rulesupCount* is the number of cases in D that contain the *condset* and are labeled with class *y*. A general rule from each *ruleitem* is *condset* $\rightarrow$ *y*. The support and confidence of this rule are computed in the same way of an association rule. An example of a *ruleitem* is <{(A,1), (B,1)}, (class, 1)>. From this *ruleitem*, A and B are attributes. *Ruleitems* that satisfy *minsup* are called *frequent ruleitems*.

The CBA-RG algorithm is based on the Apriori algorithm to find all *ruleitems* that have support above *minsup*. Let *k-ruleitem* denote a *ruleitem* whose *condset* has *k* items. Let F_k denote the set of frequent *k-ruleitem*. Each element of this set has a form <(*condset*, *condsupCount*), (*y*, *rulesupCount*)>. Let C_k be a set of candidate *k-ruleitems*. Figure 2.3 shows the CBA-RG algorithm.

```
1   F1 = {large 1-ruleitems};
2   CAR1 = genRules(F1);
3   prCAR1 = pruneRules(CAR1);
4   for (k = 2; Fk-1 ≠ ∅; k++) do
5       Ck = candidateGen(Fk-1);
6       for each data case d ∈ D do
7           Cd = ruleSubset(Ck, d);
8           for each candidate c ∈ Cd do
9               c.condsupCount++;
10              if d.class = c.class then c.rulesupCount++
11          end
12      end
13      Fk = {c ∈ Ck | c.rulesupCount ≥ minsup};
14      CARk = genRules(Fk);
15      prCARk = pruneRules(CARk);
16  end
17  CARs = ⋃k CARk;
18  prCARs = ⋃k prCARk;
```

Figure 2.3: The CBA-RG algorithm (Liu, Hsu and Ma, 1998).

The first scan over the data of the CBA-RG algorithm is represented at lines 1-3. This step (line 1) counts the item and class occurrences to determine the frequent 1-*ruleitems*. Then a set of CARs called CAR_1 is generated by *genRules* function

using the set of 1-*ruleitems* (line 2). At line 3, CAR_1 is pruned with function *pruneRules*. The function *pruneRules* uses the pessimistic error rate based pruning method in C4.5 (Quinlan, 1992). If rule *r*'s pessimistic error rate is higher than the pessimistic error rate of rule $\bar{r}$ obtained by deleting one condition from the conditions of *r*, then rule *r* is pruned.

For each pass k, there are four main operations. The first operation is to generate the candidate *ruleitems* C_k from frequent *ruleitems* F_{k-1} by the *candidateGen* function (line 5). Second, the algorithm scans the database and updates various support counts of the candidates in C_k (lines 6-12). After these new frequent *ruleitems* have been determined to form F_k (line 13), the algorithm then produces the rules CAR_k using the *genRules* function (line 14). For the last operation, these rules are pruned in line 15.

2.2.2.2 The CBA-CB Algorithm

For the CBA-CB algorithm, the set of CARs (or prCARs) from the CBA-RG algorithm is used to construct a classifier. The CBA-CB algorithm is shown in Figure 2.4.

```
1   R = sort(R);
2   for each rule r ∈ R in sequence do
3       temp = ∅;
4       for each case d ∈ D do
5           if d satisfies the conditions of r then
6               store d.id in temp and mark r if it correctly
                    classifies d;
7       if r is marked then
8           insert r at the end of C;
9           delete all the cases with the ids in temp from D;
10          selecting a default class for the current C;
11          compute the total number of errors of C;
12      end
13  end
14  Find the first rule p in C with the lowest total number
        of errors and drop all the rules after p in C;
15  Add the default class associated with p to end of C,
        and return C (our classifier).
```

Figure 2.4: The CBA-CB algorithm (Liu, Hsu and Ma, 1998).

Let *R* be the set of generated rules which are CARs or pCARs, and *D* be the training data. The concept of the algorithm is to choose a set of high precedence rules in *R* to cover *D*. There are three steps of the CBA-CB algorithm.

Step 1: This step ranks the set of generated rule *R* in decreasing order according to the precedent relation (line 1). Given two rules, r_i and r_j. r_i has a higher precedence than r_j if;

1. the confidence of r_i is greater than that of r_j, or
2. their confidences are the same, but the support of r_i is greater than r_j, or
3. both the confidences and supports of r_i and r_j are the same, but r_i is generated before r_j

Step 2: This step selects the sorted rules from the previous step to construct the classifier (lines 2-13). For each of rule *r*, *D* is scanned to find the cases covered by *r*. The rule *r* is marked if it correctly classifies a case *d*. If *r* can correctly classify at least one case, it will be a potential rule in a classifier. The cases covered by rule *r* are then removed from *D*. After that, the majority class in the remaining data is selected to be a default class. Finally, the algorithm computes and records the total number of errors classified by all rules in current classifier *C* and the default class with the training data. The rule selection process is terminated when there is no rule or no training case left.

Step 3: This step removes the rules in *C* that do not improve the accuracy of the classifier (lines 14-15). First, the algorithm finds the cutoff rule which is the first rule at which there is the least number of errors recorded on *D*. Then all rules after the cutoff rule can be discarded. Finally, the remaining rules and the default class in *C* are used to form a classifier.

In classifying an unseen case, the case is predicted as a class by the consequence of the first rule covering the case. The default class is used to classify when no covering rules in the classifier can be used.

2.2.3 Support vector machine (SVM)

A support vector machine (SVM) is a supervised learning algorithm first introduced by Vapnik (Vapnik, 1998). This algorithm can be used for classification and regression problems, but in this thesis we will focus on the classification problem. The concept of SVM is to map input vectors to a higher dimensional space and try to find a maximal separating hyperplane. Two parallel hyperplanes are constructed on each side of the hyperplane that separates the data between two classes. "The separating hyperplane is the hyperplane that maximizes the distance between the two parallel hyperplanes. An assumption is made that the larger the margin or distance between these parallel hyperplanes the better the generalization error of the classifier will be" (Wikipedia, 2008).

Let *D* be a training dataset containing labeled input vectors (x_i, y_i) where x_i is a sample data and y_i is its label, $x_i \in R^N$ and $y_i \in \{-1,1\}$ for $i=1,\ldots,m$. In a learning step, the SVM algorithm finds the hyperplane that satisfies Equation (1). Where w is a normal vector to hyperplane, $|b|/\|w\|$ is the perpendicular distance from the hyperplane to the origin, $\|w\|$ is the Euclidean norm of w, and b is the bias.

$$w \cdot x + b = 0 \tag{1}$$

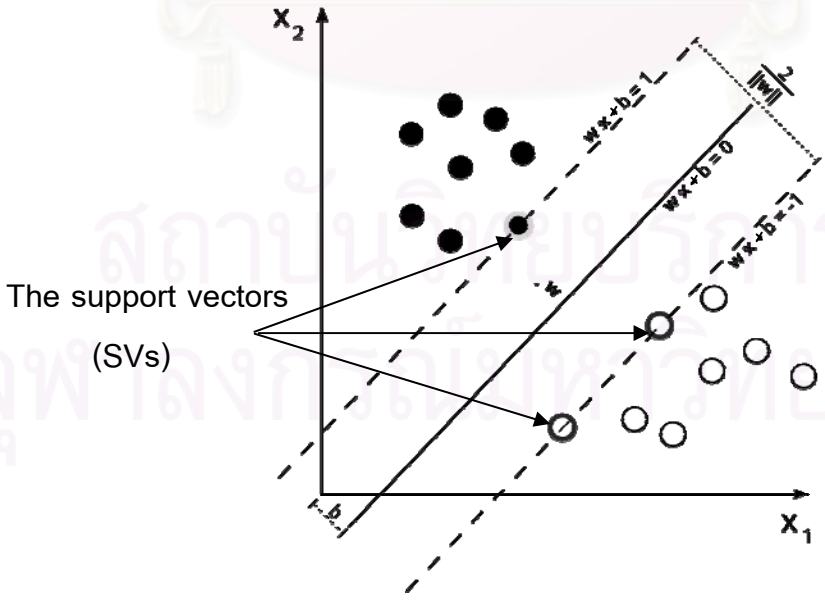


Figure 2.5: Linear separating hyperplanes (Wikipedia, 2008).

Figure 2.5 illustrates the maximum-margin hyperplane and margins for SVM trained with training data from two classes. At least one vector which two parallel hyperplanes pass through are called support vectors (SVs), or we can say samples on the margin are the support vectors.

Let d_+ (d_-) be the shortest distance from the separating hyperplane to the closest positive (negative) example. Define $d_+ + d_-$ be the margin of a separating hyperplane. For the linearly separable case, the algorithm looks for the separating hyperplane with largest margin. This can be formulated as follows. Suppose that all the training data satisfy the following constraints:

$$x_i \cdot w + b \geq +1 \quad , y_i = +1 \tag{2}$$

$$x_i \cdot w + b \leq -1 \quad , y_i = -1 \tag{3}$$

This can be combined into one set of inequalities:

$$y_i (x_i \cdot w + b) - 1 \geq 0 \quad \forall i \tag{4}$$

Consider the points that lie on the parallel hyperplanes. The point lines on the hyperplane $x_i \cdot w + b = 1$ with normal w and perpendicular distance from the origin $|1-b|/\|w\|$. Similarly, the point that lies on the hyperplane $x_i \cdot w + b = -1$ has normal w, and the perpendicular distance from the origin is $|-1-b|/\|w\|$. Hence $d_+ = d_- = 1/\|w\|$ and the margin is $2/\|w\|$. Note that two hyperplanes have the same normal (since they are parallel) and that no training points fall between them. Thus these two hyperplanes which give the maximum margin can be found by minimizing $\|w\|^2$, subject to constraints in Equation (4) (Burges, 1998).

SVM uses the function in Equation (5) to classify a new sample x. The sample x is classified as positive if $f(x) > 0$ and classified as negative if $f(x) < 0$.

$$f(x) = w \cdot x + b \tag{5}$$

In some cases, there exists no hyperplane that can separate the two classes of training data. To handle with this problem, the *Soft Margin* method (Cortes and Vapnik, 1995) is used to choose a hyperplane that splits the examples as cleanly as

possible, while still maximizing the distance to the nearest cleanly split examples. The concept of this method is to relax the constraints (2) and (3) by adding positive slack variables ξ_i as shown in Equations (6)-(7).

$$x_i \cdot w + b \geq +1 - \xi_i, \; y_i = +1 \tag{6}$$

$$x_i \cdot w + b \leq -1 + \xi_i, \; y_i = -1 \tag{7}$$

$$\xi_i \geq 0 \quad \forall i \tag{8}$$

$\sum_i \xi_i$ is an upper bound on the number of training errors. Thus the objective function to be minimized is changed to Equation (9), where C is a free parameter determined by a user. A larger C corresponds to assigning a higher penalty to errors in classifying the training data.

$$\|w\|^2 + C(\sum_i \xi_i) \tag{9}$$

In real world problems, most of the applications are non-linearly separable. To handle this problem, kernel function is used to map the input space into a higher dimensional feature space (Boser, Guyon and Vapnik, 1992). Then the algorithm constructs a maximum margin hyperplane in the high-dimensional feature space. The first kernels investigated for a pattern recognition problem are shown in Table 2.1.

Table 2.1: Summary of the non-linear kernels.

Type of support vector machine	Kernel function $K(x,y)$	Comment
Polynomial degree p	$(x.y+1)^p$	Power p is specified by the user
Gaussian radial-basis function	$e^{-\|x-y\|^2/2\sigma^2}$	The width σ^2, common to all the kernels, is specified by the user
Two layer sigmoidal neural network	$\tanh(\beta_0 x.y - \beta_1)$	For some (not every) both β_0 and $\beta_1 > 0$

2.2.4 Redial Basis Function (RBF) Network

The Redial Basis Function (RBF) network is an approach for function approximation that is closely related to distance-weighted regression and also to artificial neural networks (Powell, 1987, Broomhead and Lowe, 1988, Moody and Darken, 1989). The construction of the traditional RBF network involves three layers with entirely different roles as illustrated in Figure 2.6.

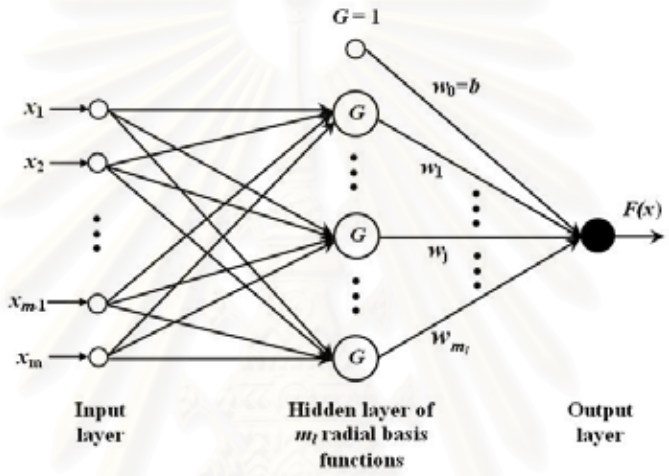


Figure 2.6: The RBF network.

As shown in Figure 2.6, the RBF network consists of three layers (Haykin, 1999). The first layer is composed of input nodes whose number is equal to the dimension of the input vector. The second layer is a hidden layer. This layer consists of nonlinear units that are connected directly to all of the nodes in the input layer. In this layer, the input space is nonlinear transformed to the hidden space. The activation functions of the individual hidden units are defined by Gaussian functions. The output layer consists of a single linear combination unit, being fully connected to the hidden layer. In this approach, the value of the output unit is a function given in Equation (10).

$$F(x) = w_0 + \sum_{i=1}^{m_1} w_i G(\|x - t_i\|) \qquad (10)$$

Where m_1 is the number of centers, vector t represents the center points, vector w is the weights in the output layer, and G is the Gaussian function (see Figure 2.6) as shown in Equations (13) and (14).

In training step, the weight vector w in the output layer of the network will be calculated by matrix computation as shown in Equation (11).

$$w = G^{+} d \tag{11}$$

Where G^{+} is the pseudo inverse of matrix G defined in equation (12) and d is the desired response vector in the training set.

$$G^{+} = (G^{T} G)^{-1} G^{T} \tag{12}$$

$$G = \{g_{ji}\} \tag{13}$$

$$g_{ji} = \exp\left(- \frac{\left\| x_j - t_i \right\|^2}{2\sigma_i^{\,2}} \right) \tag{14}$$

where i, j=1,2,…,m_1, x_j is the j th input vector of the training sample and t_i is the ith vector of the center and σ denotes the width of the Gaussian function.

There are two main approaches to specifying the centers of the radial basis functions in the hidden layer of the RBF network. The first approach assigns each training data as a radial basis function. This method is efficient in the application that does not have a large number of training data. Each of these radial basis functions may be assigned the same width σ^2. For this approach, the RBF network learns a global approximation to the target function in which each training example can influence the value of $\hat{f}$ only in the neighborhood of x_i. One advantage of this approach is that it allows the RBF network to fit the training exactly.

The second approach tries to select the set of the radial basis functions that is smaller that the number of training data. This approach is much more efficient than the first approach, especially when the number of training examples is large. The set of centers may be distributed with centers spaced uniformly throughout the total input space. A hybrid learning process is also used to find appropriate center locations.

One popular technique is a clustering algorithm which allocates one radial basis function for each cluster center.

2.2.5 *k*-Nearest Neighbor (*k*-NN)

k-Nearest Neighbor (*k*-NN) is a classic instance-based learning technique (Mitchell, 1997). This technique constructs a different approximation to the target function for each distinct query instance depending on its nearest neighbors. The *k*-NN algorithm has an assumption that all instances correspond to points in the *n*-dimension space.

Define a feature vector of an instance x to be a form $<a_1(x), a_2(x),\ldots, a_n(x)>$, where $a_r(x)$ denotes a value of the rth attribute of the instance x, and n represents the total number of attributes. The *k*-NN algorithm measures a distance between the instances x and its neighbors by using Euclidean distance. A distance between two instances x_i and x_j is defined as $d(x_i, x_j)$ calculated by Equation (15).

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2} \qquad (15)$$

Training algorithm:

- For each training example $<x, f(x)>$, add the example to the list *training_examples*

Classification algorithm:

- Given a query instance x_q to be classified,
  - Let $x_1 \ldots x_k$ denote the k instances from *training_examples* that are nearest to x_q
  - Return

$$\hat{f}(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} \delta(v, f(x_i))$$

where $\delta(a,b)=1$ if $a = b$ and $\delta(a,b) = 0$ otherwise.

Figure 2.7: The *k*-Nearest Neighbor algorithm.

Figure 2.7 shows a process of the *k*-NN algorithm. This algorithm considers discrete-valued target in a form $f : \Re^n \rightarrow V$. Let *V* be a finite set of all discrete targets (or classes). The *k*-NN algorithm assigns the target $\hat{f}(x_q)$ following the most common value of *f* among the *k* training examples nearest to x_q. For example, if *k*=1, the 1-NN algorithm assigns the $\hat{f}(x_q)$ value to the value of $f(x_i)$, where x_i is the training examples nearest to x_q . If *k* is larger than 1, the algorithm assigns the most common value among the *k* nearest training examples.

Figure 2.8 illustrates the concept of the *k*-NN algorithm with all instances are transformed into points in a two-dimensional space. A set of positive and negative training examples are shown by '+' and '-' respectively. x_q represents a query point. From this figure, the 1-NN algorithm classifies x_q as a positive class whereas the 5-NN algorithm assigns it as a negative class.

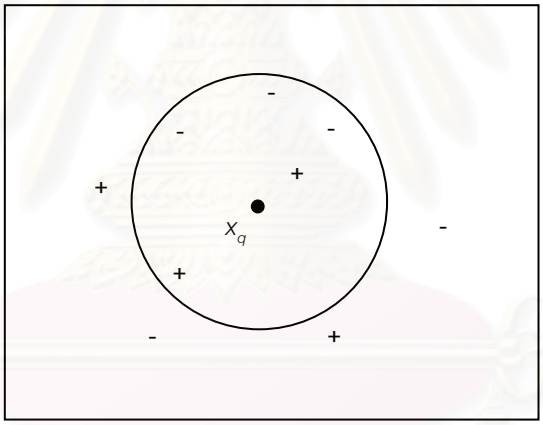


Figure 2.8: *k*-NN concept for two-dimensional space of data.

Besides discrete value of the target function, the k-NN algorithm can handle with the continuous value of the target function. To do that, the algorithm calculate the mean value of the *k* nearest training instances instead of using the most common value of the nearest examples. Thus the approximate a continuous value of the target function is performed in Equation (16).

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} f(x_i)}{k} \qquad (16)$$

One obvious modification of kNN is the distance-weighted nearest neighbor algorithm. This algorithm is more effective and widely used than the traditional k-NN algorithm. This is because the distance-weighted *k*-NN is robust to noisy training data and quite effective when it is applied to a large set of training data. The main idea of this technique is to weight the contribution of each of *k* neighbors according to its distance to the query point x_q (giving higher weight to closer neighbors). The distance-weight is calculated by Equation (17).

$$w_i = \frac{1}{d(x_q, x_i)+1} \tag{17}$$

where x_i for i =1,..., k are the k nearest training instances and $d(x_q, x_i)$ is the Euclidean distance. So, the target function value is re-defined in Equation (18).

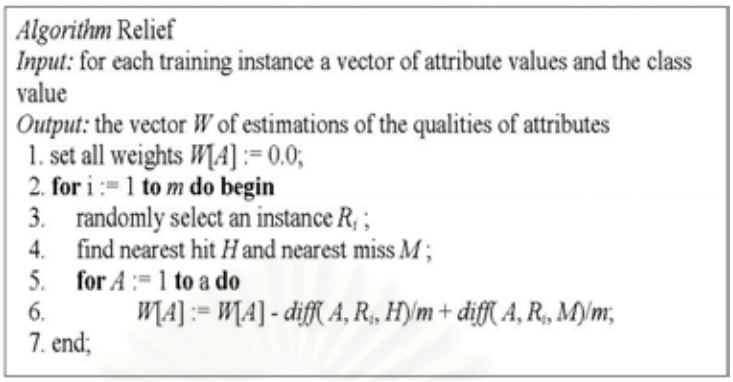
$$\hat{f}(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} w_i \delta(v, f(x_i)) \tag{18}$$

2.3 Relief Algorithms

Relief algorithms are heuristic measures for estimating the quality of the attributes. Since the original Relief algorithm (Kira and Rendell, 1992) cannot deal with incomplete and noisy data, and is limited to two class classification problems, there are many extensions of the Relief algorithm that improve the performance of the Relief algorithm. ReliefF is one of the most successful algorithm (Kononenko, 1994), that is more robust and can deal with noisy and incomplete data. Furthermore, it can deal with multiple class problems. RReliefF, an extension of ReliefF, can deal with continuous class problems (Robnik-Sikonja and Kononenko, 1997). The algorithms of Relief, ReliefF, and RReliefF are described in Sections 2.3.1, 2.3.2, and 2.3.3 respectively.

2.3.1. Relief

The main idea of Relief is to estimate the weight of each attribute according to how well its value distinguishes between instances that are near each other. The algorithm of Relief is shown in Figure 2.9.



Algorithm Relief
Input: for each training instance a vector of attribute values and the class value
Output: the vector W of estimations of the qualities of attributes
1. set all weights $W[A] := 0.0$;
2. **for** i := 1 **to** m **do begin**
3. randomly select an instance R_i;
4. find nearest hit H and nearest miss M;
5. **for** A := 1 **to** a **do**
6. $W[A] := W[A] - diff(A, R_i, H)/m + diff(A, R_i, M)/m$;
7. end;

Figure 2.9: The Relief algorithm (Robnik and Kononenko, 1997).

Relief starts with m random instances from all of the instances in the dataset. For each randomly selected instance R_i, Relief finds the nearest instance from the same class, called nearest hit H, and the nearest one from the different class, called nearest miss M. For updating the weight of attribute A, Relief considers the value of attribute A for R_i, M, and H as follows.

$W[A]$=P(different value of A | nearest miss) - P(different value of A | nearest hit) (19)

As shown in the formula, Relief tries to increase weight to the attributes that have different values for two instances from the different classes whereas it tries to decrease weight to the attributes that have different values for two instances with the same class.

Function $diff(A, I_1, I_2)$ in Figure 2.9 calculates the difference between the values of the attribute A for instances I_1 and I_2. Equations (20) and (21) below show the function $diff$ for nominal and numerical attributes respectively. For finding the nearest neighbor, Manhattan distance in (22) was used as a measure for calculating the distance between two instances.

$$diff(A, I_1, I_2) = \begin{cases} 0; value(A, I_1) = value(A, I_2) \\ 1; otherwise \end{cases} \quad (20)$$

$$diff(A, I_1, I_2) = \frac{|value(A, I_1) - value(A, I_2)|}{max(A) - min(A)} \quad (21)$$

$$\delta(I_1, I_2) = \sum_{i=1}^{a} diff(R_i, I_1, I_2) \quad (22)$$

2.3.2. ReliefF

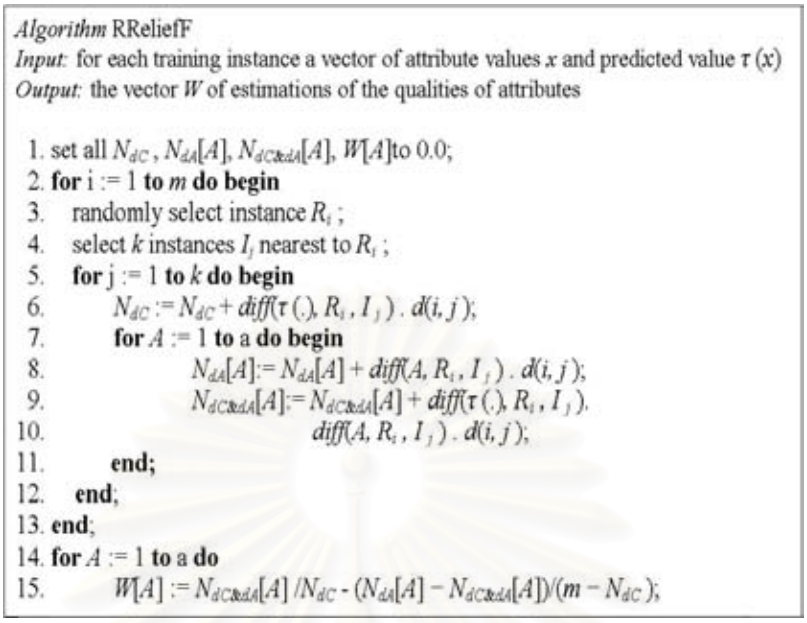
ReliefF is an extension of Relief and is more robust. ReliefF can deal with noisy data by searching for k nearest neighbors from the same class and also k nearest neighbors from the different class. To deal with multi-class problems, ReliefF updates the weight of each attribute by averaging the contribution of all the hits and all the misses as shown in lines 8 and 9 of Figure 2.10, where *a* (line 7) is a number of total attributes.

Algorithm ReliefF
Input: for each training instance a vector of attribute values and the class value
Output: the vector W of estimations of the qualities of attributes
1. set all weights $W[A] := 0.0$;
2. **for** i := 1 **to** m **do begin**
3. randomly select an instance R_i ;
4. find k nearest hits H_j ;
5. **for** each class $C \neq class(R_i)$ **do**
6. from class C find k nearest misses $M_j(C)$;
7. **for** $A := 1$ **to** a **do**
8. $W[A] := W[A] - \sum_{j=1}^{k} diff(A, R_i, H_j)/(m.k) +$
9. $\sum_{C \neq class(R_i)} \left[\frac{P(C)}{1 - P(class(R_i))} \sum_{j=1}^{k} diff(A, R_i, M_j(C))\right]/(m.k)$;
10. **end**;

Figure 2.10: The ReliefF algorithm (Robnik and Kononenko, 1997).

2.3.3 RReliefF

RReliefF, extended from ReliefF, was designed for continuous class problems. Therefore it does not find the nearest hits and misses like ReliefF, but it uses the probability of the relative distance between continuous class values of two instances to estimate the weight of the attributes. RReliefF applies Bayes' rule for calculating the weight of attribute A as shown in Equation (23). The algorithm of RReliefF is shown in Figure 2.11.



Algorithm RReliefF
Input: for each training instance a vector of attribute values x and predicted value $\tau(x)$
Output: the vector W of estimations of the qualities of attributes

1. set all N_{dC}, $N_{dA}[A]$, $N_{dC\&dA}[A]$, $W[A]$to 0.0;
2. **for** i := 1 **to** m **do begin**
3. randomly select instance R_i;
4. select k instances I_j nearest to R_i;
5. **for** j := 1 **to** k **do begin**
6. $N_{dC} := N_{dC} + diff(\tau(.), R_i, I_j) \cdot d(i,j)$;
7. **for** A := 1 **to** a **do begin**
8. $N_{dA}[A] := N_{dA}[A] + diff(A, R_i, I_j) \cdot d(i,j)$;
9. $N_{dC\&dA}[A] := N_{dC\&dA}[A] + diff(\tau(.), R_i, I_j) \cdot$
10. $diff(A, R_i, I_j) \cdot d(i,j)$;
11. **end;**
12. **end;**
13. **end;**
14. **for** A := 1 **to** a **do**
15. $W[A] := N_{dC\&dA}[A] / N_{dC} - (N_{dA}[A] - N_{dC\&dA}[A]) / (m - N_{dC})$;

Figure 2.11: The RReliefF algorithm (Robnik and Kononenko, 1997).

$$W[A] = \frac{P_{diffC|diffA} P_{diffA}}{P_{diffC}} - \frac{(1 - P_{diffC|diffA}) P_{diffA}}{1 - P_{diffC}} \tag{23}$$

where P_{diffA} = P(different value of A | nearest instances), P_{diffC} = P(different prediction | nearest instances), and $P_{diffC|diffA}$ = P(different prediction | different value of A and nearest instances)

In Figure 2.11, $\tau(.)$ in lines 6 and 9 represents the continuous value of the prediction. N_{dC}, $N_{dA}[A]$, and $N_{dC\&dA}[A]$ represent the weights for different continuous value $\tau(.)$, different attribute, and different prediction & different attribute respectively. In addition, $d(i,j)$ is the term that calculates the influence of the distance between instances R_i and I_j.

$$d(i,j) = \frac{d_1(i,j)}{\sum_{l=1}^{k} d_1(i,l)} \tag{24}$$

where

$$d_1(i,j) = e^{-\left(\frac{rank(R_i, I_j)}{\sigma}\right)^2} \tag{25}$$

$rank(R_i,I_j)$ in Equation (25) is the rank of the instance I_j in a sequence of instances ordered by the distance from R_i and σ is a user defined parameter for controlling the influence of the distance.

2.4 Composite Classifier

The term of a composite classifier or an ensemble of classifiers is used to identify a set of classifiers whose individual decisions are combined in some way to classify new examples (Dietterich, 1997).

There are two strategies of classifier combination: classifier selection and classifier fusion (Kuncheva, 2002). For classifier fusion, it assumes that all classifiers are equally experienced in the whole feature space and the decisions of all classifiers are taken into account for classifying a new example x. The assumption of classifier selection is that each classifier has expertise in some local area of the feature space. When a feature vector $x \in \Re^n$ is submitted for classification, the classifier responsible for the neighborhood of x is given the highest authority to label x. There are two types of classifier selection methods: static and dynamic (Kuncheva, 2002). The static method proposes one best classifier for the whole data space, while the dynamic method takes into account the characteristics of a new instance to be classified.

For the composite classifier construction, there are two architectures. The first one is combining homogeneous classifiers. This method generates a composite classifier by a single algorithm. This means that all component classifiers are learnt by the same algorithm. An important requirement of this architecture is the diversity of training data. This method manipulates the training set to generate multiple classifiers. The learning algorithm runs several times, each time using a different distribution of the training examples. This technique works especially well for unstable learning algorithms. An unstable learning algorithm is the algorithm whose output provides major changes in response to small changes in the training data.

The second architecture is combining heterogeneous classifiers. This method uses different learning algorithms to form a composite classifier. An example framework of this architecture is called “Stacked Generalization” proposed by Wolpert (1992). This framework consists of two layers of the classifiers as shown in Figure 2.12.

The classifiers at the level-0 receive inputs from the original data and each classifier outputs a prediction. The classifier at the second layer receives the predictions of the level-0 classifiers as input and outputs the final prediction. The concept of Stacked Generalization is to minimize the generalization error by using the classifiers in higher layers to learn the type of errors made by the classifiers in the previous level. The level-1 classifier tries to learn how previous classifiers make mistakes in classes they agree or disagree and uses this knowledge when making predictions.

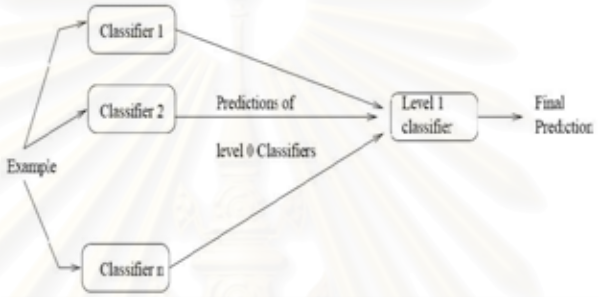


Figure 2.12: Stacked Generalization architecture (Gama, 2000).

The following subsections describe two classifier combination methods that will be used to compare with our dynamic classifier combination method. These two algorithms are examples of classifier fusion methods.

2.4.1 Majority Vote

Majority vote is the simplest and a classical method for combining classifiers. This method is implemented by counting the number of classifiers which make the predictions to each class labels. Finally, it gives the class label having the highest summation as an output.

Let $d_{i,j}$ be the prediction of the t^{th} classifier from the set of classifiers D, $d_{t,j} \in \{0,1\}, t = 1,...,T$ and $j = 1,...,c$, where T is the number of classifiers and c is the number of classes. $d_{t,j} = 1$ if the t^{th} classifier predicts class j, and $d_{t,j} = 0$, otherwise. The vote will then result in an ensemble decision for class k if k is satisfied by Equation (26).

$$\sum_{t=1}^{T} d_{t,k} = \max_{j=1}^{c} \sum_{t=1}^{T} d_{t,j} \qquad (26)$$

2.4.2 Naïve Bayes

This method assumes that the classifiers are mutually independent. Let $D = \{D_1, D_2, \ldots, D_L\}$ be a set of classifiers and $\Omega = \{\omega_1, \ldots, \omega_c\}$ be a set of class labels. For each classifier D_j, a c x c confusion matrix CM^j is calculated by applying D_j to the training data. The element $cm^j_{k,s}$ is the number of elements of the dataset whose true class label was ω_k, and are assigned by D_j to class ω_s. Let $cm^j_{.,s}$ is the total number of samples labeled by D_j into class ω_s calculated by the summation of the sth column of CM^j. Let LM^j is a c x c probability matrix of CM^j. The element $lm^j_{k,s}$ is an estimate of the probability that the true label is ω_k given that D_j assigns crisp class label ω_s.

$$lm^j_{k,s} = P(\omega_k \mid D_j(x) = \omega_s) = \frac{cm^j_{k,s}}{cm^j_{.,s}} \tag{27}$$

Let $s_1,\ldots, s_L$ be the crisp class labels assigned to a new instance x by classifiers $D_1, D_2, \ldots, D_L$, respectively. By the independence assumption, the estimate of the probability that the true class label is ω_i, is calculated by Equation (28), where i=1, …, c.

$$\mu_i(x) \propto \prod_{j=1}^{L} P(\omega_i \mid D_j(x) = s_j) = \prod_{j=1}^{L} lm^j_{i,s_j} \tag{28}$$

Finally, this method assigns the class label to instance x according to the maximum probability of the true class label.

2.5 Related Works

At present, there are many techniques for genotypic HIV-1 drug resistance prediction. The following contents are the literature reviews of the application of HIV-1 drug resistance prediction.

For the genotypic HIV-1 drug resistance interpretation application, many systems use rule-based techniques (Shafer, Jung and Betts, 2000, Meynard, ray, Morand, et al, 2002, Laethem, Luca, Antinori, et al, 2002, Reid, Bassett, Day, et al, 2002). These systems contain the rules encoding information from the medical literature as the knowledge base. One of these tools, the HIVdb system, is an online genotypic HIV-1 resistance interpretation system constructed by Stanford University (Shafer, Jung and

Betts, 2000). This system uses the mutation scoring tables to calculate a score from each sequence and interprets drug susceptibility into one of five classes ranging from susceptible to high-level resistant. However, there are some limitations of the HIVdb system such as the sensitivity to the drug cannot always be deduced from the viral nucleotide sequence due to high polymorphism and limited knowledge of the role of interaction among these amino acid substitutions (Rhafer, Jung and Betts, 2000, Rhee, Gonzales, Kantor, Betts, Revela and Shafer, 2003).

Besides genotypic resistance interpretation systems, a variety of techniques have been applied to phenotypic drug resistance from genotype such as statistical analysis, and machine learning techniques. The phenotypic results from these techniques are classified into two or more classes of drug susceptibility depending on the certain cutoff values.

For statistical analysis, multiple linear regression analysis (REG) was applied to the construction of a separate regression model for each drug (Wang, Jenwitheesuk, Samudrala and Mitter, 2004). In the model, the dependent variable is the logarithm of the IC50 fold change, while the independent variables are dummy variables corresponding to mutations. In addition, this technique uses the stepwise regression method to optimize the parameters for each independent variable. Moreover, cluster analysis, recursive partitioning, and linear discriminant analysis are used to investigate the relationship between results of genotypic and drug susceptibility phenotypic assays (Sevin, DeGruttola, Nijhuis, Schapiro, Foulkes, Para, and Boucher, 2000).

Machine learning is the most popular approach applied to the prediction of phenotype resistance from genotype. Geno2Pheno is the online system used to predict the phenotypic resistance. This system constructed the model using the support vector machines (SVMs). At the beginning (Beerenwinkel, et al., 2001), the system focused on binary classification: susceptible or resistant. In this system, linear kernel was used to map an input space into a feature space. In 2003 (Beerenwinkel, et al., 2003), the Geno2Pheno system was developed to SVM regression models. Still, the system was constructed with the linear kernel with an epsilon-insensitive loss function.

Other supervised learning algorithms have been used to deal with this problem such as decision trees (Beerenwinkel, et al., 2001, 2002) and artificial neural networks (ANNs) (Wang and Larder, 2003). These algorithms classify drug susceptibility into one of two classes: susceptible or resistant. Furthermore, the self-organizing map, an unsupervised learning algorithm, was used to classify drug susceptibility into one of three classes: high, medium, or low resistant (Draghici, S. and Potter, 2003). Most of the works mentioned above use a single algorithm to classify drug resistance.

During the recent years, many bioinformatics applications applied ensemble classifiers to construct the model for the classification tasks. Most of them used a single learning algorithm to construct the ensemble classifiers and combined the final prediction with the majority voting algorithm. In 2006, Shen and Chou (2006) proposed the ensemble classifier for protein fold pattern recognition. The ensemble of this work was formed by a set of base classifiers, each of which was trained by different parameters. The individual classifiers were optimized evidence-theoretic *k* nearest neighbors (OET-*k*NN) rules. The final prediction was combined by the weighted voting algorithm. In the same year, Stepenosky, et al. (2006) presented the ensemble of three multilayer perceptron (MLP) classifiers combined with majority vote and decision templates method for an early diagnosis of Alzheimer's disease.

In 2007, Liu, Zhu and Feng (2007) developed the ensemble classifier by fusing ten basic individual K-local Hyperplane Distance Nearest Neighbor (HKNN) classifiers through majority voting scheme. Recently, Tsymbal, et al. (2008) proposed an ensemble learning approach for the antibiotic resistance prediction. In this work, a set of classifiers were built over different time periods. Each base classifier is given a weight proportional to its local accuracy with regard to the instance tested, and the best base classifier is selected, or the classifiers are integrated using weighted voting.

CHAPTER III

SINGLE CLASSIFIERS CONSTRUCTION

This chapter describes the procedure used to construct various models by using a single classifier, i.e. CBA, SVM, the RBF network, and *k*-NN. For single classifiers construction, there are four steps which are described in more detail in the following subsections.

3.1 Initial Data Collection

In the first step, all pairs of genotype-phenotype data for 6 drugs of Protease Inhibitors (PIs) which are LPV, APV, NFV, IDV, SQV and RTV, 6 drugs of Nucleoside Reverse Transcriptase Inhibitors (NRTIs) which are 3TC, ABC, AZT, D4T, ddC and ddI, and 3 drugs of Non-Nucleoside Reverse Transcriptase Inhibitors (NNRTIs) which are DLV, EFV and NVP were downloaded from Stanford HIV RT and Protease Sequence Database with the ViroLogic Susceptibility test method. Table 3.1 shows an example of HIV-1 protease resistance database with primary and secondary amino acid substitutions that are different from the HIV-1 wild-type strain. A capital letter appeared in each column represents amino acid which is different from the HIV-1 wild-type.

After the database was downloaded, all genotype data were transformed to sequences of amino acid by comparing with HIV-1 reference strain pNL4-3. Table 3.2 shows an example of amino acid sequences of the HIV-1 protease genes and its phenotypic results for the NFV drug. The last column of this table represents the class of drug susceptibility: susceptible (S) and resistant (R).

There are 99 amino acid positions from position 1 to position 99 in protease gene (or PI drug) whereas there are 201 amino acid positions from position 40 to position 201 in reverse transcriptase gene (or NRTI and NNRTI drugs). The total samples, percentage of susceptible (S) and resistant (R) classes, and the cutoff value for each drug are shown in Table 3.3. The phenotypic results were assigned into one of two classes: susceptible or resistant according to the cutoff value of each drug.

Table 3.1: HIV-1 protease resistance database with primary and secondary amino acid substitutions that are different from the HIV-1 wild-type strain (pNL4-3).

Sample ID	HIV-1 Protease amino acid position																					
	10	20	24	30	32	33	36	46	47	48	50	53	54	63	71	73	77	82	84	88	90	93
CA12529	I	-	-	-	-	-	-	-	-	-	-	-	-	P	-	-	-	-	-	-	-	L
CA12878	-	-	-	-	-	-	I	-	-	-	-	-	-	P	-	-	-	-	-	-	-	-
CA13194	H	-	-	-	-	-	-	I	-	-	-	-	-	P	V	-	I		-	-	M	L
P649	I	-	-	-	-	-	-		-	-	-	-	-	A	-	-	-	-	-	-	-	-
RC-1080.1	I	I	-	-	-	I	I		-	-	-	-	V	P	V	-	-	T	-	-	M	-
RC-1211.2	I	R	-	-	-	-	I		-	-	-	-	V	P	-	-	-	A	-	-	M	-
SD-10	-	I	-	-	-	-	-	-	-	-	-	-	-	P	-	-	I	-	-	-	-	L
SD-15_1	V	-	-	-	-	-	-	I	-	-	-	-	-	P	-	-	I	-	-	-	-	L
SD-19	I	-	I	-	-	-	-	L	-	-	-	-	V	P	V	-	-	A	-	-	-	-
SD-2	-	-	-	-	-	-	-	-	-	-	-	-	-	P	-	-	-	-	-	-	-	-

Table 3.2: The examples of genotype-phenotype data.

Sample ID	HIV-1 Protease amino acid position																											Class
	1	...	10	...	20	...	24	...	30	...	36	...	46	...	54	...	63	...	71	...	82	...	90	...	93	...	99	
CA12529	P		I		K		L		D		M		M		I		P		A		V		L		L		F	S
CA12878	P		L		K		L		D		I		M		I		P		A		V		L		I		F	S
CA13194	P		H		K		L		D		M		I		I		P		V		V		M		L		F	R
P649	P		I		K		L		D		M		M		I		A		A		V		L		I		F	S
RC-1080.1	P		I		I		L		D		I		M		V		P		V		T		M		I		F	R
RC-1211.2	P		I		R		L		D		I		M		V		P		A		A		M		I		F	R
SD-10	P		L		I		L		D		M		M		I		P		A		V		L		L		F	S
SD-15_1	P		V		K		L		D		M		I		I		P		A		V		L		L		F	R
SD-19	P		I		K		I		D		M		L		V		P		V		A		L		I		F	R
SD-2	P		L		K		L		D		M		M		I		P		A		V		L		I		F	S

Table 3.3: Detail of total datasets.

Drug	Total Samples	Percent of Susceptible Class	Percent of Resistant Class	Cutoff Value
LPV	319	52	48	10.0
APV	541	55	45	2.0
NFV	626	33	67	2.5
IDV	595	45	55	2.1
SQV	606	50	50	1.7
RTV	573	57	53	2.5
3TC	529	32	68	3.5
ABC	529	59	41	4.5
AZT	528	43	57	1.9
d4T	530	61	39	1.7
ddC	394	44	56	1.7
ddI	528	68	32	1.7
DLV	554	65	35	2.5
EFV	563	62	38	2.5
NVP	577	56	44	2.5

3.2 Feature Subset Selection

Since total amino acid positions of the HIV-1 protease gene and reverse transcriptase gene are 99 and 201 respectively and some of them are irrelevant or redundant, these attributes may decrease the performance of the learning algorithm. To alleviate this problem, a feature selection technique is used to select important attributes from the training data. Besides improving the predictive accuracy, selecting the important attributes also reduces learning and testing times of the models.

In this paper, RReliefF, a classical feature estimation algorithm, was used to select important attributes for each drug (Robnik-Sikonja and Kononenko, 1997). RReliefF is an extended version of ReliefF which has been used to select important

attributes in many applications in medical areas (Hilario, et al., 2004, Luts, et al., 2007, Huang, et al., 2007). Whereas ReliefF is designed for handling data with a discrete class, RReliefF is able to deal with data with a continuous class as in our case where phenotypic drug resistance is a real value (continuous class). Though, the final prediction of our method is discrete classification (resistant or susceptible), we found from experiments that RReliefF provides more accurate results in the classification than ReliefF. Therefore, RReliefF is used in our method.

After applying RReliefF to the training data, we selected the attributes, which have the weights higher than or equal to θ where θ was set to 0.01. We set the threshold of RReleifF to 0.01 because this threshold provides the number of selected attributes close to the number of selected attributes by a rule-based method used by Stanford HIV Drug Resistance Database. Moreover, amino acid positions (attributes) selected by RReliefF and those recommended by Stanford HIV Drug Resistance Database share several common attributes.

The relations between amino acid positions which were selected by rule-based and RReliefF methods for each drug are illustrated in Figure 3.1 – Figure 3.3. These figures show the number of attributes that were selected by the rule-based and RReliefF methods. In addition, the number in the intersection between two cycles shows the number of common attributes of the two methods for each drug. The percentage of the intersection between rule-based and RReliefF methods is computed by the proportion of the number of common attributes between two methods to the total number of attributes selected by the rule-based method.

The relation in Figure 3.1 shows that the set of attributes selected by RReliefF for each PI drug shared some attributes with the rule-based method and the percentage of these common attributes was higher than or equal to 55.0%. For NRTIs drugs (see Figure 3.2), the percentage of the intersection was higher than 70.0% for all drugs except for 3TC (44.44%) and DDC (41.18%). The percentage of the intersection for all NNRTIs drug (see Figure 3.3) was greater than 86.0%. However, in our experiments, the attributes selected by RReleifF were different for each fold depending on its training data.

Though, for the same drug, mutation positions in different HIV genotypes can be different, some of them may not affect the drug susceptibility. RReleifF has ability to select only important mutation positions from the training set that are considered to be able to distinguish between instances with the susceptible class and instances with the resistant class.

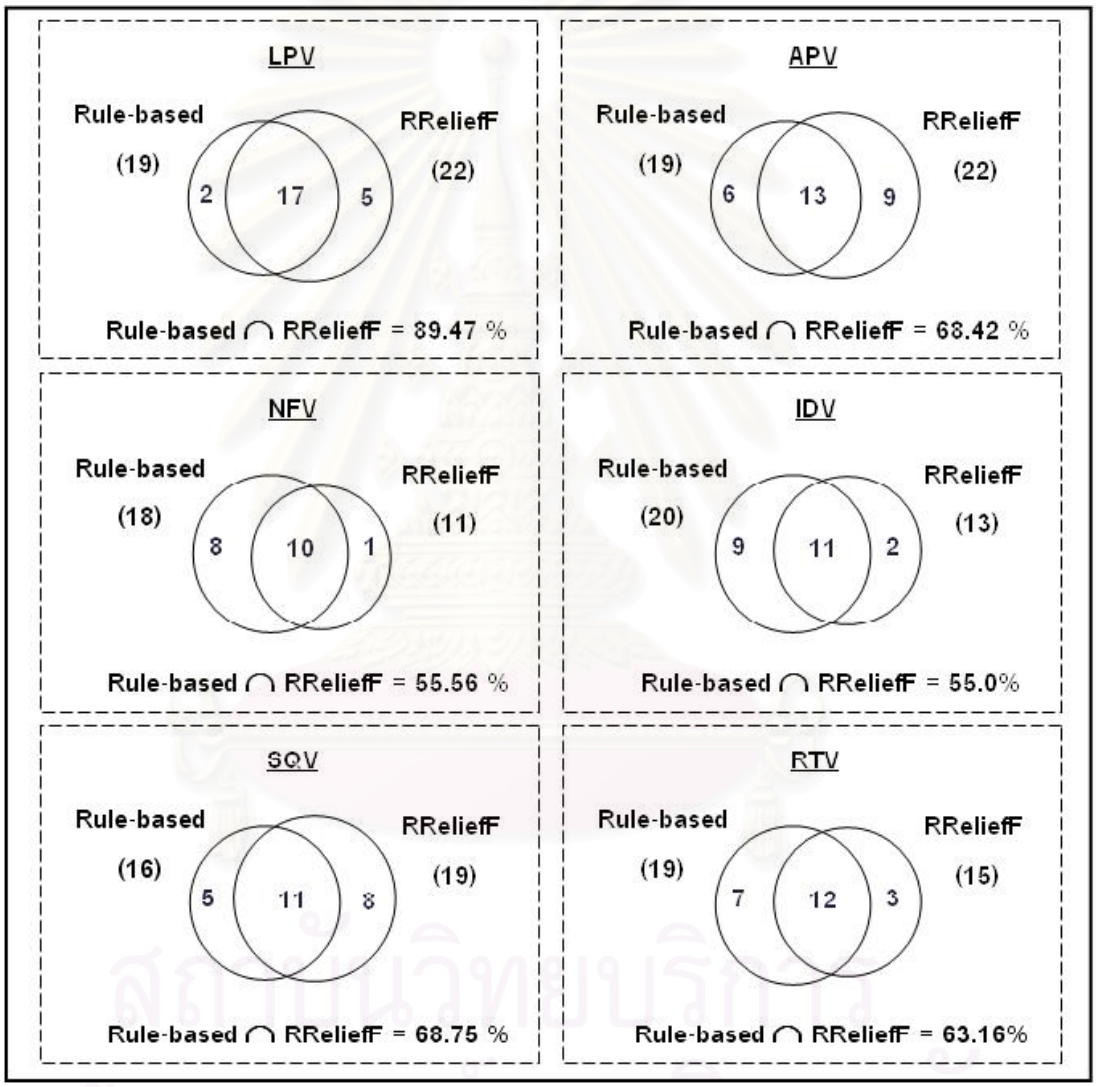


Figure 3.1: Relations between the attributes selected by the rule-based and RReliefF methods for PIs drugs.

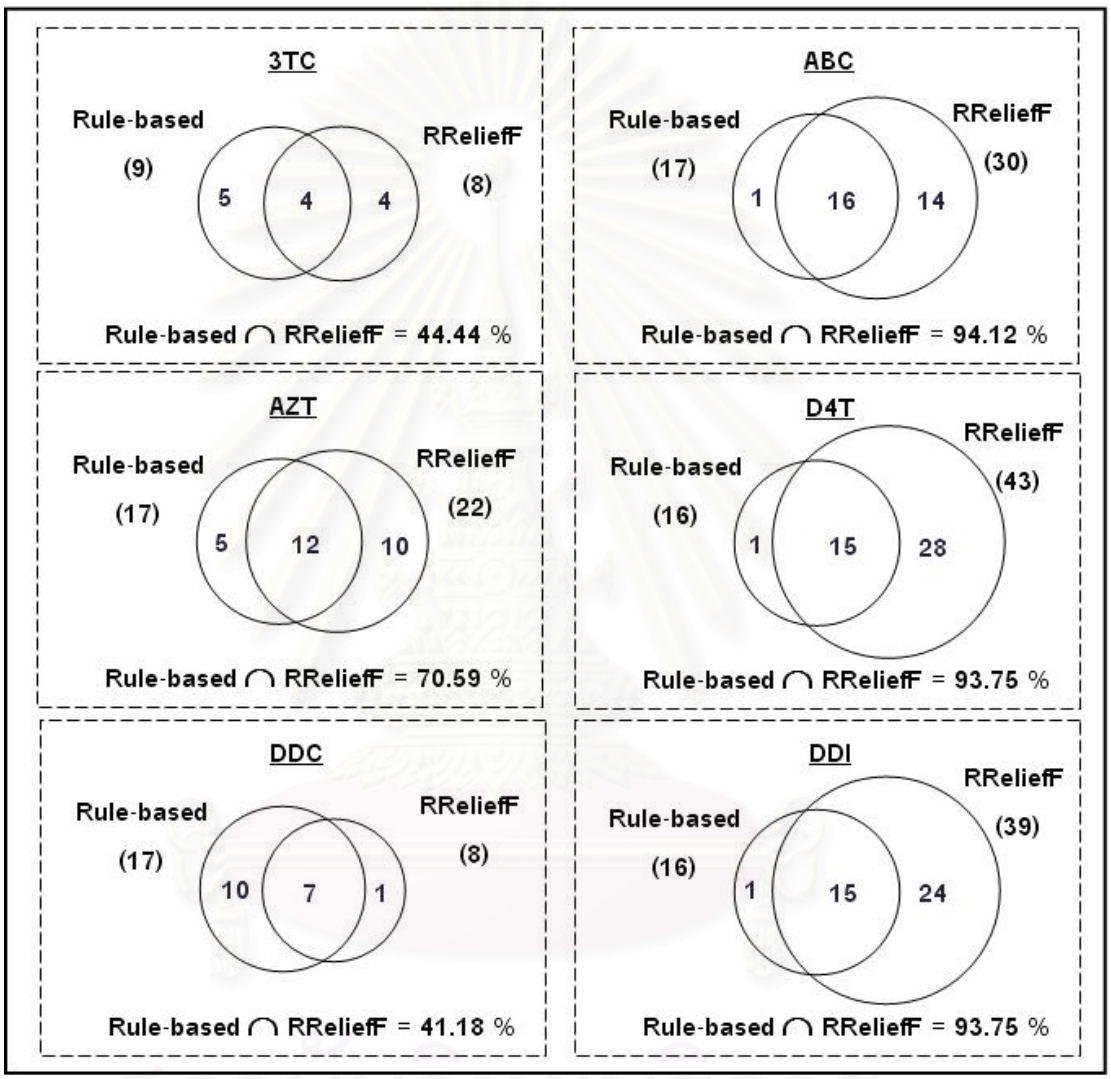


Figure 3.2: Relations between the attributes selected by the rule-based and RReliefF methods for NRITs drugs.

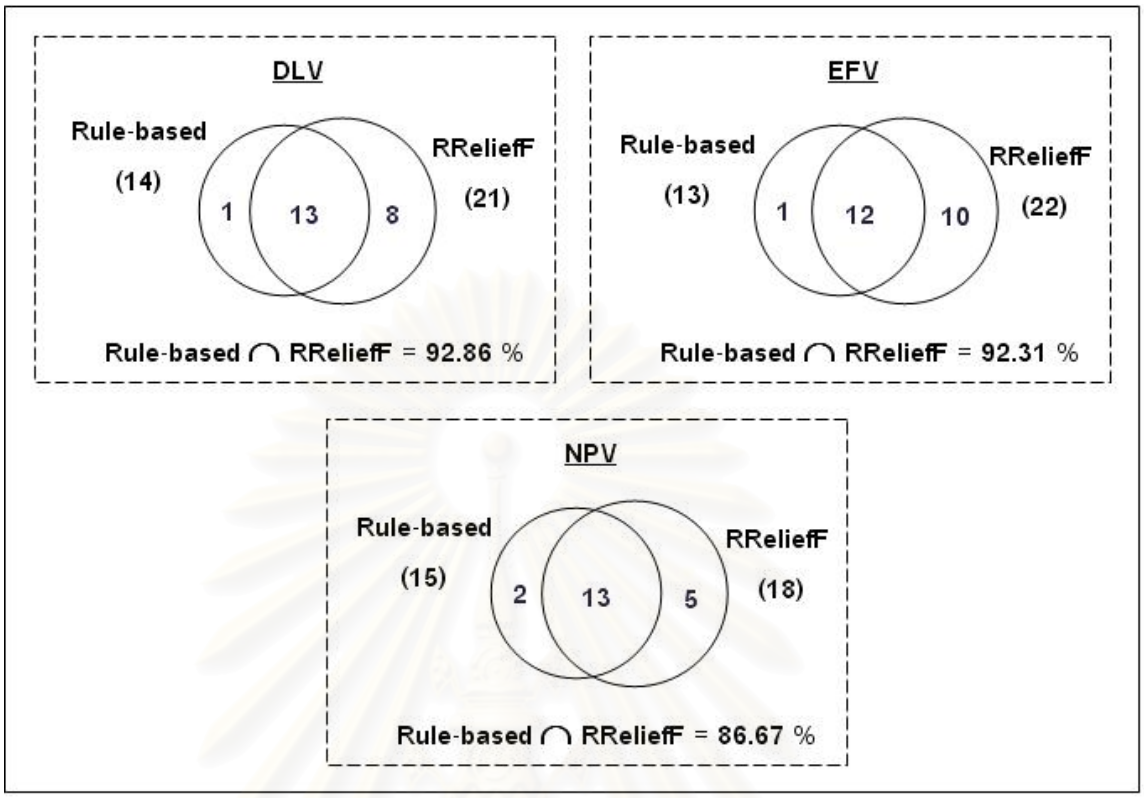


Figure 3.3: Relations between the attributes selected by the rule-based and RReliefF methods for NNRITs drugs.

3.3 Data Transformation

Since genotype data are represented as amino acid sequences, these data have to be transformed into a suitable format for a learning algorithm. Several approaches to the description of protein sequences have been proposed. For example, the knowledge of the hydropathy blocks is used to translate the protein sequence to a fixed-dimensional vector (Huang, Zhao, Huang and Cheung, 2006). For a protein secondary structure prediction problem, all amino acid sequences are converted to real number matrices by using a position specific scoring matrix algorithm (Jones,1999, Ghosh and Parai, 2008). Moreover, the string kernel-based method such as spectrum, mismatch, and wildcard kernels are applied for SVM (Davis, Hawkins, Maetschke and Bodén, 2006). In data transformation process of this work, there are two steps to

construct an input vector for SVM, the RBF network and *k*-NN from amino acid sequence data.

In the first step, a sequence of amino acid positions was transformed into a binary vector. Each amino acid position provided 20 binary input dimensions (there were 20 amino acids which might occur in any position). As there were 99 and 201 amino acid positions of the HIV-1 protease (PR) gene and reverse transcriptase (RT) gene respectively, at the beginning the number of input attributes of PR and RT were 1980 and 4020 respectively. However, after feature selection process, the number of input attributes for each drug was reduced significantly. This is because RReliefF eliminated irrelevant attributes and selected important attributes in the feature selection process.

After a sequence of amino acid positions has been transformed into a binary vector, each binary element in the vector was assigned a weight. In assigning the weight, RReliefF was used again to estimate the weight of each attribute (binary attribute from binary vector). Finally, the final input value of element *i* in the input vector was defined by Equation (29).

$$A_i = \begin{cases} 1 + w_i & if \ \ B_i = 1 \\ 0 & if \ \ B_i = 0 \end{cases} \tag{29}$$

Where A_i was the final input value, B_i was the value of element *i* in the binary vector, and w_i was the weight (a real value between -1 and 1) from RReliefF of attribute *i*.

Thus, at the end of this process, the inputs of each drug were real-valued vectors whose dimensions depended on the number of attributes selected by RReliefF.

3.4 Model Construction

For the model construction process, CBA, SVM, the RBF network, and *k*-NN were used to construct the classifiers separately for each drug. In the experiments, we used the same training and testing datasets for all learning algorithm. However, the input formats of CBA and other algorithms were different. The input of the CBA classifier was a sequence of amino acids while the inputs of the SVM, RBF network and *k*-NN classifiers were the real-valued vectors as described in Section 3.3. The output of

CBA, SVM, and *k*-NN was y={-1,1} where -1 (1) represents the susceptible (resistant) class whereas the output of the RBF network was a real value representing a logarithm of IC50 fold change.

In the process of CBA model construction, we set the maximum length of generating rules to 5. In addition, the pruning technique was used to reduce the number of rules without losing the prediction accuracy. The *minconf* was set to 100% while *minsup* was tuning in the range of 1% to 30%. In this experiment, several subsets of attributes selected by various feature subset selection methods were used to construct the models. The sets of selected attributes that yielded the best performance were used as the inputs for the other learning algorithms including the composite classifiers.

For SVM, several kernel functions which were linear, polynomial degree 2, 3, and 4, and RBF were used to map an input space into a feature space. These kernels used the same *C* of Equation (9) in Section 2.2.3 in learning the models. For the RBF kernel, we varied the width of the RBF function in the range of 0.01 to 30.

For constructing the models using the RBF network, each training example was represented as a center in the hidden layer and σ for each center was set to the same value. Thus the number of hidden nodes was equal to the number of total training examples. To evaluate the predictive performance of the RBF network, an output from the model was classified to the susceptible or resistant class using the cutoff value (as shown in Table 3.3). In the experiments, we varied σ values in the range of 1.0 to 3.5.

For *k*-NN, the class label of a new instance was assigned by the distance weighted *k*-NN. In the experiment, we set the number of *k* to 1, 3, 5, 7, and 9. The number that yielded the highest average accuracy was used to construct the base classifiers for the composite classifiers.

The CBA models were constructed by DMII-CBA, a data mining tool developed at School of Computing, National University of Singapore. SVM, the RBF network, and *k*-NN techniques were implemented by Matlab and SVM toolbox Version 2.51. To evaluate the performance of four single classifiers in the testing process, 10-fold cross-validation technique was used. For selecting parameters of CBA (*minsup*), SVM and the RBF network (the width of RBF), we chose the same parameter values for

each drug. To find the suitable parameters, each data fold was further divided to 5-fold, then 5-fold cross-validation was used to measure the predictive performance of each parameter values. The training and testing data of 10-fold and 5-fold cross-validation are shown in Figure 3.4.

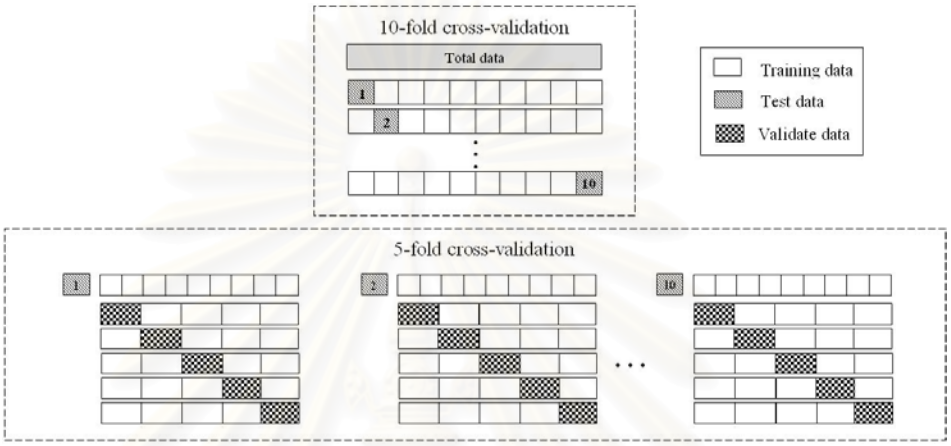


Figure 3.4: Training and testing data for single classifier construction.

CHAPTER IV

COMPOSITE CLASSIFIERS CONSTRUCTION

This chapter focuses on the composite classifiers construction. At first, the criteria to select the component classifiers are described. Then the proposed composite classifier combination approach is presented in the last section.

4.1 Composite Classifier Construction Criteria

There are three main criteria for designing a composite classifier: accuracy, diversity of the component classifiers, and efficiency of the entire composite classifier (Skalak, 1997).

The accuracy of the component classifiers is the most important criteria. If the predictions that are being combined are not highly accurate, then the ultimate prediction accuracy will be difficult to be achieved. Hansen and Salamon (1990) demonstrated that the composite classifier is most useful when its component classifiers make errors independently with respect to others. They proved that when all the component classifiers have the same error rate and that the error is less than 0.5 with the assumption that their errors are completely independently, the expected ensemble error must decrease monotonically with the number of classifiers. On the other hand, if the error rate is more than 0.55, the error rate of the composite classifier is monotonically increased.

The diversity of the component classifiers is a necessary factor in classifier combination. Ali and Pazzani (1996) have shown that error is mostly reduced by using component classifiers whose errors are low correlated. In this work, we use error correlation which has been implemented by Ali and Pazzani (1996) as a measure of diversity of component classifiers. This measure compares the output of the components with the correct target class.

Let $f(x) = S_i$ denote that instance x belongs to class S_i, and $\hat{f}_i(x) = S_i$ mean that the classifier $\hat{f}_i$ predicts class S_i for instance x. The definition of the error correlation is the probability that two component classifiers make the same error as shown in Equation (30) (Gama, 2000).

$$\phi_{ij} = p(\hat{f}_i(x) = \hat{f}_j(x) \mid \hat{f}_i(x) \neq f(x) \vee \hat{f}_j(x) \neq f(x)) \quad (30)$$

For the efficiency of the composite classifier, we consider using a small number of component classifiers. Given equal performance, one would prefer smaller component classifiers because it takes less time in training and application. Some research indicated that a small number of classifiers can be enhanced the accuracy of the composite classifier. For example, a handwritten digit recognition system (Battiti and Collar 1994) used only two to three neural networks to give the higher accuracy than the best from an individual network. In addition, for the weather prediction task (Kwok and Carter 1990), the experiments showed that the error rate reached a minimum with only three or fewer component decision trees.

4.2 Dynamic Classifier Combination (DCC)

Our proposed composite classifier (called DCC) is a heterogeneous architecture classifier that dynamically selects base classifiers according to each test instance and uses a classifier fusion method for combining base classifiers. The concept of DCC is to select the suitable classifiers to form the composite classifier. These classifiers are dynamically chosen by a heuristic function depending on the prediction of each base classifier in classifying a new instance x. After the base classifiers are selected, DCC uses dynamic weighted voting to classify the new instance x. The architecture of DCC is illustrated in Figure 4.1.

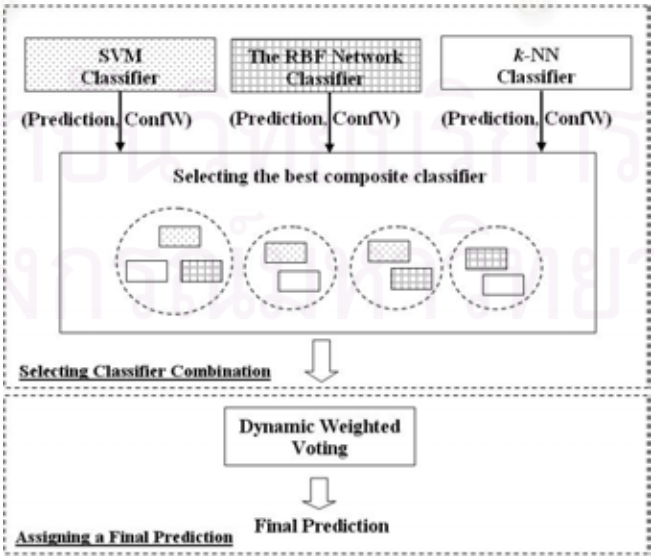


Figure 4.1: Dynamic Composite Classifier architecture.

There are two main steps of DCC: selecting classifier combination and assigning a final prediction. These steps are described in the following subsections.

4.2.1 Selecting Classifier Combination

This step dynamically selects the combination of the base classifiers for a new instance x based on a heuristic function. Let *BCC* be the set of base classifier combinations, i.e. *BCC*={{SVM, RBF network, *k*-NN}, {SVM, RBF network}, {RBF network, *k*-NN}, {SVM, *k*-NN}}. A suitable classifier combination of the new instance x is the member of *BCC* that has the maximum value of the heuristic function $cw_i(x)$. i is an index of classifier combination pattern (member of *CBB*). This function is shown in Equation (31). $meanAcc_i$, $stdAcc_i$, and $meanConfW_i$ are the average percentage of the training accuracies from the training instances near to x of the base classifiers in the i^{th} classifier combination of *BCC*, standard deviation of accuracies of that composite classifier, and the average percentage of the $ConfW$ values from the base classifiers in the i^{th} classifier combination respectively. EC_i is a value of error correlation of the i^{th} classifier combination calculated by Equation (30).

$$cw_i(x) = \frac{(meanAcc_i \times meanConfW_i - stdAcc_i)}{EC_i} \quad (31)$$

$ConfW$ measures the confidence of base classifier i in correctly classifying the new instance x into susceptible (S) or resistant (R) class. The higher value of the confidence weight implies that the base classifier has more confidence in the classification. Each base classifier has a different function for calculating $ConfW$. The Equations (32)-(34) show the formula of $ConfW$ for SVM, the RBF network, and *k*-NN respectively.

As shown in Equation (32), $dist(h, x)$ is the distance between the instance x and the separating hyperplane (h) of SVM. For the RBF network, $cutoff$ represents the cutoff value and $out(x)$ is an output value of the instance x from the RBF network. For *k*-NN, $diffW(x)$ represents the difference between the weight of prediction of class S and the weight of prediction of class R in the classification process of the weighted *k*-NN classifier. $typ(x)$ represents *typicality* of the instance x (Zhang, 1992). This value

measures the confidence of an instance x in instance-based classification. It is defined as the proportion between the average distance from the instance x to instances of different classes to the average distance from the instance x to instances of the same class.

$$ConfW_{SVM}(x) = \sqrt{dist(h, x)} \tag{32}$$

$$ConfW_{RBF\ network}(x) = \sqrt{\left|cutoff - out(x)\right|} \tag{33}$$

$$ConfW_{kNN}(x) = \left|diffW(x)\right| \times typ(x) \tag{34}$$

Note that $ConfW$ of each base classifier is normalized to [0..1] by the min-max normalization method, where the minimum and maximum values are taken from the training data.

4.2.2 Assigning a Final Prediction

After the base classifiers are selected by the previous step (these classifiers are formed to be a composite classifier), the predictive information of the base classifiers are sent to the Dynamic Weighted Voting (DWV) algorithm. This algorithm computes dynamic weights of each base classifier and predicts the final prediction using locally weight voting.

When x is fed to the composite classifier, the weight of each base classifier j (w_j) is computed by Equation (35).

$$w_j(x) = ConfW_j(x) + PerfW_j(x) \tag{35}$$

Where $ConfW_j(x)$ is the same value as shown in Equations (32)-(34) in Section 4.2.1.

The performance weight $PerfW_j$ measures the predictive performance of base classifier j in correctly classifying training instances near the new sample x. Let s be the class label predicted by base classifier j. This performance measurement employs training instances near x to estimate the local accuracy of the base classifier j with respect to class s. $PerfW_j$ is calculated by the proportion of the samples that are near to x whose true labels are s.

For assigning a final prediction of the composite classifier, DWV compares the total weight w^+ to the total weight w^-, where w^+ and w^- are the summation of weight w_i of all base classifiers i that predict class R, and the summation of those for class S, respectively. Finally, DWV outputs the final prediction for the new instance x according to the larger total weight.

4.3 Training and Evaluation Phases of the Composite Classifier

There are two phases of composite classifier construction: the training phase and the evaluation phase.

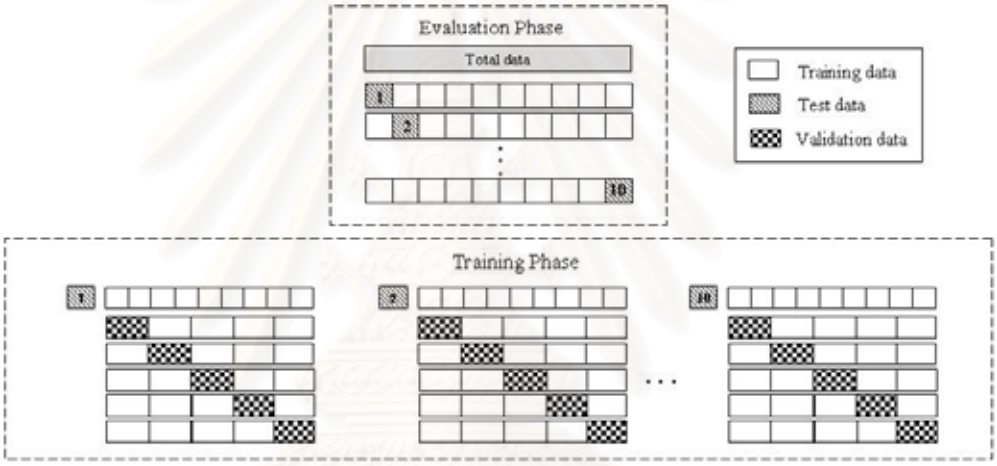


Figure 4.2: Training and testing data for composite classifier construction.

4.3.1 Training Phase

1. Train the base classifiers using training data from 5-fold cross-validation. After training, classify validation data into two classes (-1 or 1). Training and validation data are shown in Figure 4.2 in the training phase. After this step, store the predictions of each base classifier for the total training data.
2. Since in the step 1, base classifiers have not been trained on the entire training data, re-train the base classifiers on the training data in the evaluation phase of Figure 4.2.

4.3.2 Evaluation Phase

When a new example is presented (a test data in Figure 4.2 in the evaluation phase), it is classified by all base classifiers. Then the predictions of all base classifiers are sent to the Dynamic Classifier Combination (DCC) algorithm. The DCC algorithm combines suitable classifiers to form a composite classifier and predicts the final prediction.

CHAPTER V

EXPERIMENTAL RESULTS AND DISCUSSION

This chapter shows the experimental results. The content of this chapter consists of two main parts. First, the experimental results of each learning algorithm are presented. Then, we compare the predictive performance of four learning algorithms with HIVdb and Geno2Pheno systems. In addition, the predictive behaviors of each learning algorithm are analyzed in this chapter. The latter part shows the comparison of the predictive performance between the proposed classifier combination method and other methods. Then the discussion of how our proposed method enhances the predictive performance of the single classifiers is presented.

5.1 Performance Evaluation Measurement

In the experiments, 10-fold cross-validation was used to minimize the bias associated with the random sampling of the training and testing data in comparing the predictive accuracy of four learning algorithms.

In this study, three performance measures were used to evaluate the predictive performance. These measures were accuracy, sensitivity, and specificity. The sensitivity is the probability of correctly predicting a positive (resistant) sample whereas the specificity is the probability of correctly predicting a negative (susceptible) sample.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{36}$$

$$sensitivity = \frac{TP}{TP + FN} \tag{37}$$

$$specificity = \frac{TN}{TN + FP} \tag{38}$$

where *TP* denotes the number of resistant examples which are classified as resistant,

TN denotes the number of susceptible examples which are classified as susceptible,

FP denotes the number of susceptible examples which are classified as resistant, and *FN* denotes the number of resistant examples which are classified as susceptible.

5.2 Single Classifier Results and Analysis

5.2.1 The Results of CBA Models

For constructing the classifiers from CBA algorithm, the *minsup* value of each data fold was selected by 5-fold cross-validation. In this experiment, we compared the predictive performances between each set of attributes selected by different feature selection methods. In Allmutant approach, each attribute which had only one value on all transactions of each drug was eliminated. For Rule-based approach, we selected the important attributes, recommenced by Stanford HIV Drug Resistance Database. For RReliefF, we ran RReliefF to select important attributes for each drug. Note that the subset of selected attributes for each data fold was different depending on the training data.

Table 5.1: The comparisons of the predictive accuracy of each feature selection method.

Drug	Allmutant	Rule-based	RReliefF
LPV	**86.83**	84.95	85.58
APV	85.57	85.75	**86.88**
NFV	92.02	91.70	**92.97**
IDV	89.23	89.90	**93.45**
SQV	88.26	89.76	**90.26**
RTV	93.02	93.02	**94.24**
3TC	90.60	**92.63**	89.79
ABC	83.74	84.68	**85.07**
AZT	88.07	**92.05**	**92.05**
d4T	85.66	86.23	**86.60**
ddC	78.27	78.98	**83.50**
ddI	77.47	**79.17**	**79.17**
DLV	**88.45**	88.44	87.00
EFV	89.34	**92.89**	91.47
NVP	90.29	92.20	**92.55**
average	87.12	88.16	**88.71**

The results in Table 5.1 showed that the sets of attributes selected by RReliefF provided the best average accuracy. In addition RReliefF gave the highest accuracy on eleven drugs. Rule-based and Allmutant methods yielded the best accuracy on four and two drugs, respectively. From the results of this experiment, we decided to use only the set of attributes selected by RReliefF for the further experiments.

5.2.2 The Results of SVM Models

This subsection illustrates the experimental results of the SVM algorithm. Several kernel functions of SVM were run to evaluate the predictive performance for this application. The prediction results of all kernel functions are shown in Table 5.2 and Figure 5.1.

Table 5.2: The comparisons of the predictive accuracy of each kernel function.

Drug	Poly 2	Poly 3	Poly 4	Linear	RBF
LPV	56.08	53.92	50.48	86.83	**88.40**
APV	54.68	52.83	50.79	85.75	**88.17**
NFV	64.40	54.72	47.49	92.34	**93.13**
IDV	56.45	53.08	53.07	93.10	**93.45**
SQV	56.05	53.63	47.21	90.25	**90.76**
RTV	53.10	51.53	53.10	94.07	**95.46**
3TC	64.08	58.42	53.70	91.31	**91.68**
ABC	55.57	51.04	47.45	83.73	**86.58**
AZT	51.52	54.18	48.29	92.42	**93.18**
d4T	55.09	50.94	50.38	82.83	**86.04**
ddC	50.62	50.37	49.10	84.45	**84.77**
ddI	52.34	50.45	47.74	78.98	**79.17**
DLV	48.71	52.23	51.85	88.45	**90.07**
EFV	50.64	50.66	51.68	93.07	**94.32**
NVP	51.12	53.74	48.52	92.02	**92.72**
average	54.70	52.78	50.06	88.64	**89.86**

Column Poly 2, Poly 3, or Poly 4 in Table 5.2 represents the polynomial kernel function degree 2, 3, or 4 respectively. The experimental results in Table 5.2 indicate that the suitable kernel function for the application of HIV-1 drug resistance prediction was the RBF kernel since it provided the best predictive performance for all drugs. The second best kernel is the linear kernel. The predictive results of polynomial kernel functions show that this type of kernel function was not suitable for this application. In addition, the higher degree of polynomial function, yielded the lower predictive performance. Since the RBF kernel function gave the best predictivie performance for this application, we used this kernel to construct SVM classifiers for the composite classifiers. The overview of all kernel functions is illustrated in Figure 5.1.

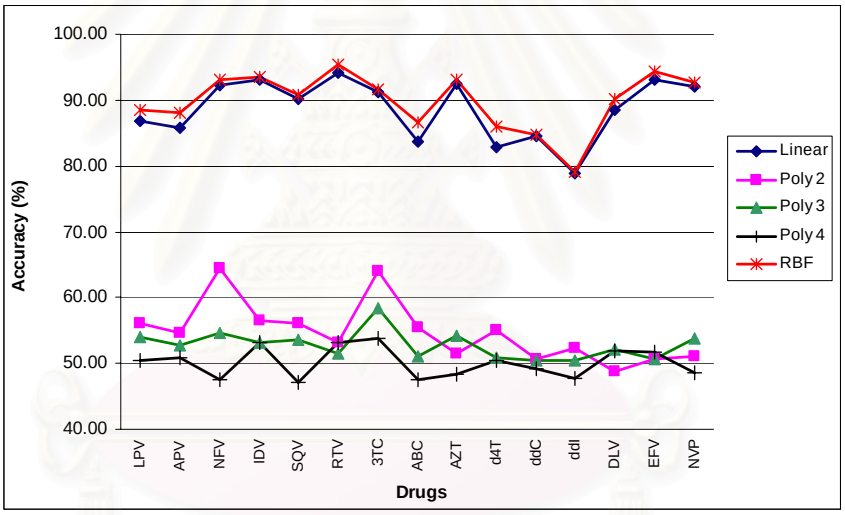


Figure 5.1: The predictive accuracy of each kernel function.

5.2.3 The Results of RBF Network Models

From all our experiments, this is the only one algorithm that outputs continuous values. These outputs represent the logarithm of fold change for each drug. Figures 5.2-5.4 show the overview of the comparison between target function and predictive function generated by the RBF network models. Note that all graphs in Figures 5.2-5.4 were constructed from the testing data belonging to only one folder.

The results from Figures 5.2-5.4 demonstrate that for most of drugs, the prediction of the RBF network models were consistent with the target functions especially for ddC and EFV drugs.

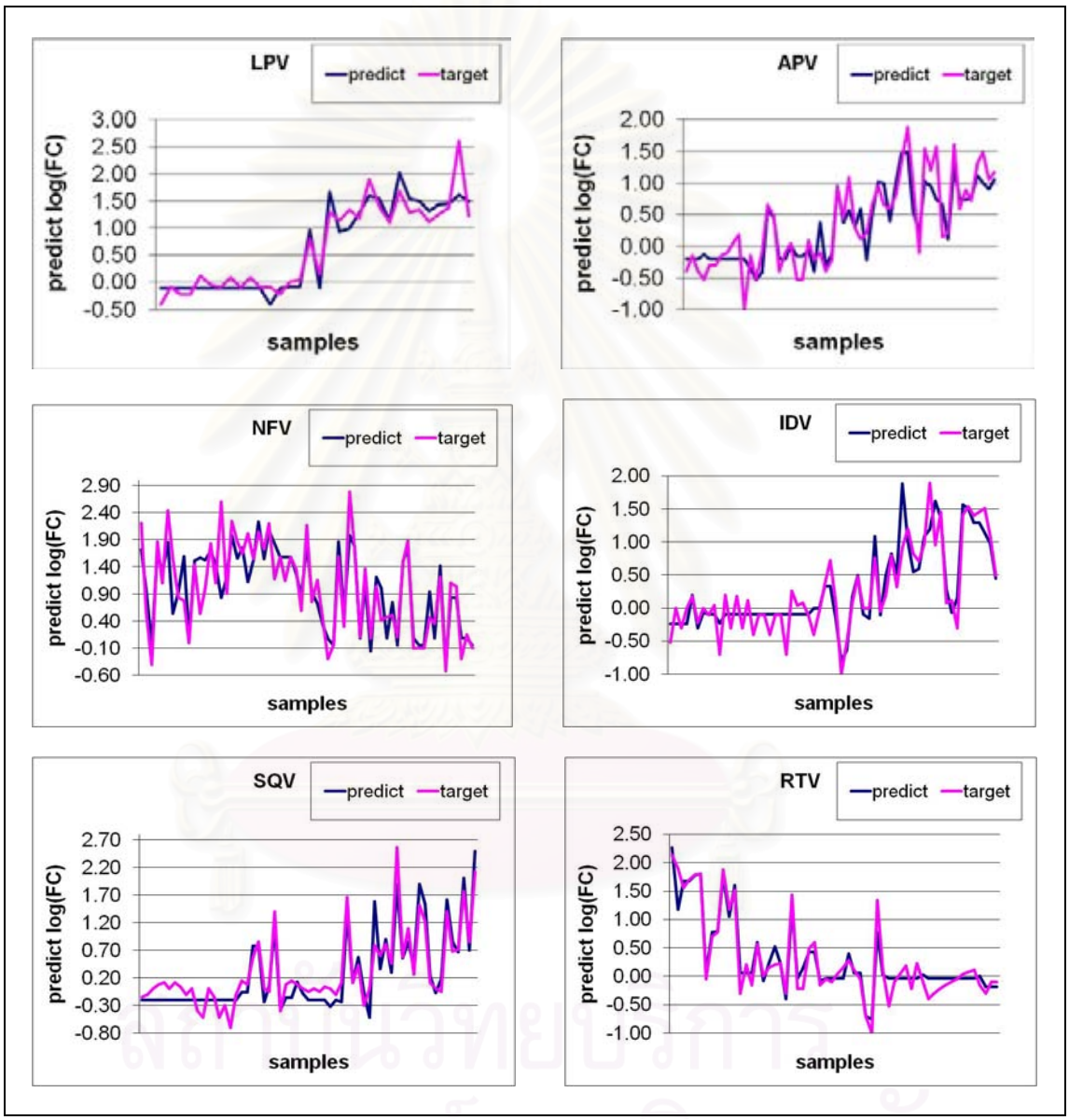


Figure 5.2: The comparison graphs between the target function and the predictive function for PIs drugs.

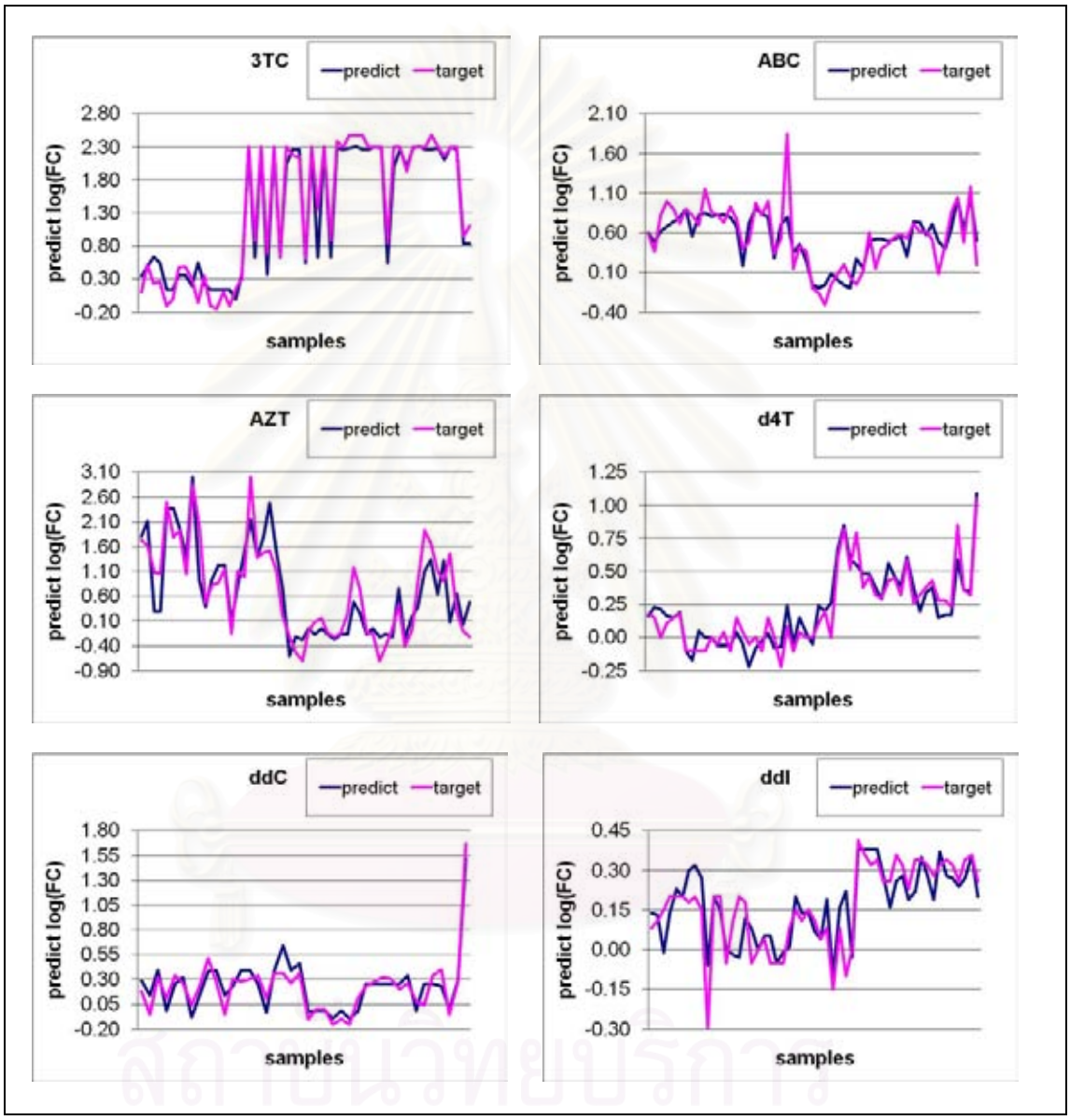


Figure 5.3: The comparison graphs between the target function and the predictive function for NRTIs drugs.

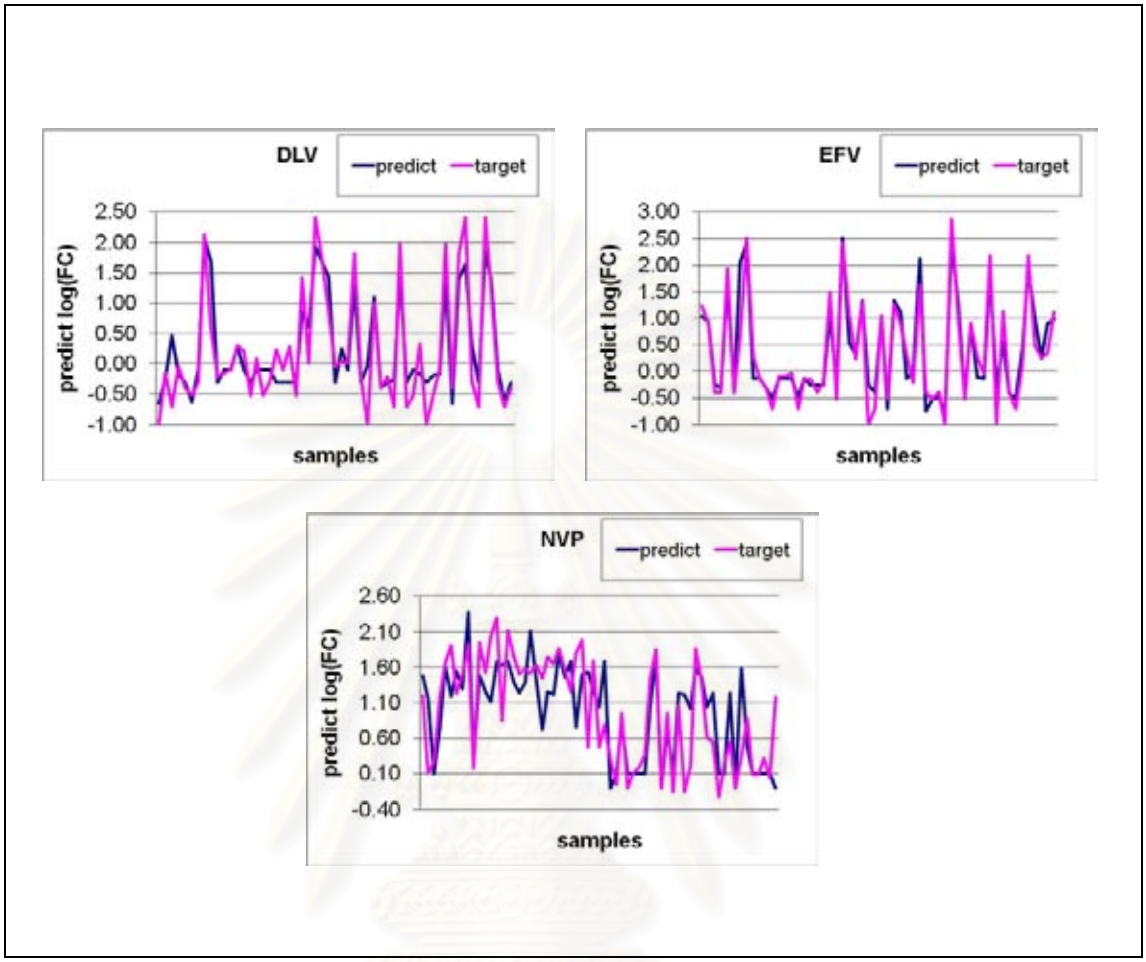


Figure 5.4: The comparison graphs between the target function and the predictive function for NNRTIs drugs.

5.2.4 The Results of *k*-NN Models

In *k*-NN model construction, several numbers of *k* were run to compare the predictive performance. Table 5.3 shows the predictive results of *k*-NN models with different numbers of *k*. From the experimental results, we found that predictive accuracies of models increased with the increase of *k*. Thus we selected *k*=9 for constructing *k*-NN base classifiers for the composite classifier.

Table 5.3: The accuracy of *k*-NN models when *k* is varied.

Drug	1-NN	3-NN	5-NN	7-NN	9-NN
LPV	82.76	82.76	84.33	86.21	**87.46**
APV	83.55	83.55	83.92	84.10	**84.66**
NFV	90.26	88.50	91.69	92.01	**92.17**
IDV	89.58	90.08	91.60	**92.61**	92.44
SQV	87.29	87.29	**89.11**	88.12	87.62
RTV	92.15	92.15	**92.85**	92.32	92.15
3TC	88.85	90.36	91.12	**91.49**	**91.49**
ABC	83.55	83.74	**84.31**	83.74	83.74
AZT	91.10	91.10	91.48	**91.86**	91.67
d4T	82.45	82.45	83.96	84.91	**86.23**
ddC	63.45	81.47	82.74	83.00	**83.76**
ddI	78.03	78.03	79.55	**80.87**	80.30
DLV	86.64	86.64	**88.63**	87.91	88.09
EFV	88.10	89.70	**92.19**	90.76	90.59
NVP	91.16	91.16	**91.16**	**91.16**	90.30
average	85.26	86.60	87.91	88.07	**88.18**

5.2.5 The Comparisons of Four Single Classifiers

The results in Table 5.4 show the percentage of the sensitivity and specificity of four algorithms. The results showed that all algorithms except for the RBF network provided an average specificity value higher than sensitivity especially for *k*-NN while the RBF network had an average sensitivity value higher than specificity. However, when comparing the sensitivity and specificity of four algorithms, we found that the RBF network had the highest average sensitivity whereas *k*-NN provided the highest average specificity. These results indicated that the RBF network had the highest ability to correctly classify positive (resistant) examples and *k*-NN had the best performance in correctly classifying negative (susceptible) examples.

Table 5.4: The sensitivity and specificity of four single classifiers.

Drug	Sensitivity (%)				Specificity (%)			
	SVM	RBF network	*k*-NN	CBA	SVM	RBF network	*k*-NN	CBA
LPV	88.24	**89.54**	83.01	83.01	88.55	88.55	**91.57**	87.95
APV	86.94	**90.20**	80.00	84.90	**89.19**	86.15	88.51	88.51
NFV	94.96	94.96	93.77	**95.44**	**89.47**	86.12	89.00	88.04
IDV	**95.14**	94.53	92.40	93.92	91.35	88.72	92.48	**92.86**
SQV	91.81	**92.15**	84.30	91.13	89.78	86.90	**90.74**	89.46
RTV	**97.37**	94.74	89.47	94.41	93.31	93.31	**95.17**	94.05
3TC	**94.48**	**94.48**	**94.48**	88.12	85.63	82.04	85.03	**93.41**
ABC	84.40	**88.53**	73.85	81.19	88.10	85.53	**90.68**	87.78
AZT	95.70	**96.03**	94.70	94.04	**89.82**	84.07	87.61	89.38
d4T	81.34	**88.04**	79.43	85.65	89.10	85.67	**90.65**	87.23
ddC	**94.55**	93.18	92.73	93.18	**72.41**	70.69	**72.41**	71.26
ddI	60.82	**80.12**	48.54	59.65	87.96	80.67	**95.52**	88.52
DLV	81.63	**82.65**	75.00	78.06	94.69	92.18	**95.25**	91.90
EFV	91.20	**94.44**	80.09	88.43	96.25	95.10	**97.12**	93.37
NVP	87.06	**90.59**	80.39	86.67	97.21	95.03	**98.14**	97.21
average	88.38	**90.95**	82.81	86.52	89.52	86.72	**90.66**	89.40

As shown in Table 5.5, compared to the accuracy of four learning algorithms, SVM gave the highest average accuracy and had the highest accuracy on nine drugs. SVM yielded the accuracy between 79.17% (for ddI) to 95.46% (for RTV) whereas *k*-NN provided the worst average accuracy (80.30% for ddI to 92.44% for IDV). The RBF network yielded the second best of the highest average accuracy, and had the best accuracy on six drugs. The accuracy of the RBF network was between 80.49% (for ddI) to 94.85 (for EFV). The accuracy of CBA was between 79.17% (for ddI) to 94.24% (for RTV). The accuracy of each data fold for four learning algorithms are shown in Tables A.1-A.4 in Section A.1 of Appendix A.

Note that the accuracy of ddC for the HIVdb system cannot be measured because the HIVdb system did not have ddC drug for testing. When comparing the accuracy of four learning algorithms with Geno2Pheno and HIVdb systems, we found that all learning algorithms provided the higher average accuracy than two online systems. In addition, the learning algorithms yielded the best accuracy on all drug except for NFV DLV and NVP on which the HIVdb system gave the best accuracy.

Table 5.5: The comparison of the predictive accuracy for all classifiers.

Drug	Accuracy (%)					
	SVM	RBF Network	*k*-NN	CBA	Geno2pheno	HIVdb
LPV	88.40	**89.03**	87.46	85.58	81.51	73.98
APV	**88.17**	87.99	84.66	86.88	85.77	85.58
NFV	93.13	92.01	92.17	92.97	88.18	**93.93**
IDV	**93.45**	91.93	92.44	**93.45**	90.59	92.27
SQV	**90.76**	89.44	87.62	90.26	85.31	86.96
RTV	**95.46**	94.07	92.15	94.24	91.97	94.07
3TC	**91.68**	90.55	91.49	89.79	86.01	91.12
ABC	86.58	**86.77**	83.74	85.07	78.45	73.16
AZT	**93.18**	90.91	91.67	92.05	89.21	91.86
d4T	86.04	**86.60**	86.23	**86.60**	67.74	78.11
ddC	**84.77**	83.25	83.76	83.50	61.17	-
ddI	79.17	**80.49**	80.30	79.17	75.00	67.99
DLV	90.07	88.81	88.09	87.00	88.99	**91.16**
EFV	94.32	**94.85**	90.59	91.47	91.47	93.96
NVP	92.72	93.07	90.30	92.55	90.64	**93.93**
average	**89.86**	89.32	88.18	88.71	83.47	86.29

5.2.6 Data Analysis

The distribution of data is an important factor on the predictive performance of the learning algorithm. If the distribution of susceptible and resistant samples in the datasets is known, this information may help us explain why the predictive accuracies of learning algorithms (SVM, the RBF network, *k*-NN, and CBA) in classifying HIV-1 drug resistance are different.

In this subsection, we analyze the distribution of susceptible and resistant samples for each drug using hierarchical clustering implemented with Matlab. In running hierarchical clustering, we used the real-value vectors from total samples as inputs to the clustering algorithm, and set the number of clusters to thirty clusters for all drugs. After we ran the clustering algorithm, each sample in a cluster was assigned its actual class label in order to view the distribution of susceptible and resistant samples.

The information in Table 5.6 shows the number of susceptible samples (in column S) and resistant samples (in column R) for each cluster. This information shows the distribution of susceptible and resistant samples in each drug. Considering the clusters for all drugs, we found that all drugs always had only one cluster that contained susceptible samples greater than 75% of total susceptible samples, and there were eleven drugs that contained susceptible samples greater than 90% of total susceptible samples in one cluster. In addition, the number of clusters which contained susceptible samples was less than the number of clusters which contained resistant samples significantly. Thus we can conclude that in most of datasets the distribution of susceptible samples was tight and the distribution of resistant samples was scattered.

Table 5.6: The number of susceptible (S) and resistant (R) samples in the clusters for all drugs.

Cluster	LPV		APV		NFV		IDV		SQV		RTV		3TC		ABC		AZT		D4T		DDC		DDI		DLV		EFV		NVP	
	S	R	S	R	S	R	S	R	S	R	S	R	S	R	S	R	S	R	S	R	S	R	S	R	S	R	S	R	S	R
1	0	7	14	55	0	65	0	3	1	3	268	99	5	0	264	158	0	2	1	31	0	4	0	2	0	9	0	1	0	1
2	1	2	6	2	0	4	0	2	0	2	1	9	0	1	2	0	0	1	0	1	0	1	0	5	0	1	0	1	0	2
3	0	1	1	6	203	118	262	158	15	57	0	5	0	6	33	20	2	0	1	1	1	1	340	134	346	133	329	158	0	2
4	0	1	1	2	0	5	0	9	7	31	0	1	2	1	3	1	0	1	0	1	5	0	1	3	0	8	2	0	1	0
5	1	2	0	6	1	38	0	4	285	136	0	5	127	32	1	0	0	3	300	113	163	187	1	1	0	4	5	7	1	6
6	0	2	0	2	0	4	0	1	0	1	0	30	3	3	1	2	0	2	0	1	0	1	1	1	1	1	2	8	0	4
7	6	61	265	116	0	2	3	92	0	1	0	89	0	1	0	1	220	238	1	4	0	2	1	0	0	4	3	1	316	182
8	1	1	0	1	2	81	1	5	0	1	0	3	0	2	3	0	0	14	1	6	1	2	6	3	1	0	0	1	0	13
9	147	30	4	6	0	6	0	1	0	1	0	4	0	1	0	2	0	1	0	1	0	1	2	0	0	3	0	2	0	3
10	3	2	2	10	0	3	0	4	0	1	0	3	0	1	0	7	0	1	12	16	0	1	1	1	0	4	2	0	0	3
11	0	1	0	2	0	2	0	1	0	1	0	1	4	1	0	3	1	0	0	1	0	1	0	1	0	1	0	1	0	4
12	0	1	0	5	0	1	0	3	0	3	0	1	22	7	1	2	0	1	0	1	0	1	2	0	1	2	0	1	0	1
13	1	0	0	3	0	4	0	2	0	2	0	16	2	267	0	3	0	2	0	3	1	0	0	2	1	4	1	2	0	1
14	1	11	0	1	0	1	0	2	0	6	0	1	0	3	0	1	0	6	0	1	0	1	0	1	0	4	0	2	0	1
15	1	5	0	5	0	1	0	1	0	2	0	7	1	1	0	1	0	5	0	15	0	1	0	1	1	0	0	2	0	11
16	2	6	1	1	0	46	0	2	0	3	0	1	0	2	0	1	0	5	0	1	0	1	0	1	0	1	1	1	1	4
17	1	3	0	4	0	3	0	2	1	2	0	3	0	1	1	2	0	1	0	1	1	0	0	2	0	1	0	1	0	2
18	1	2	1	0	0	1	0	4	1	7	0	4	0	6	0	1	0	1	1	1	0	1	1	0	2	0	1	0	0	1
19	0	1	1	0	0	2	0	1	0	2	0	7	0	2	1	0	0	3	2	0	0	1	0	1	0	2	0	3	0	1
20	0	1	0	1	0	3	0	3	0	13	0	3	0	2	0	1	1	1	1	0	1	2	0	1	0	1	0	1	0	1
21	0	1	0	1	1	0	0	3	0	2	0	1	1	7	1	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1
22	0	3	0	3	0	9	0	1	0	3	0	2	0	5	0	1	0	1	0	1	0	1	0	2	3	3	1	2	2	0
23	0	2	0	3	0	2	0	15	1	5	0	1	0	2	0	1	0	1	0	1	0	1	0	2	0	1	0	7	1	0
24	0	1	0	1	0	1	0	3	0	3	0	1	0	1	0	1	0	1	0	1	0	2	1	0	0	1	0	1	0	1
25	0	1	0	2	1	8	0	2	2	0	0	2	0	1	0	2	0	3	0	1	1	0	0	1	1	0	0	1	0	2
26	0	1	0	2	0	3	0	1	0	1	0	1	0	1	0	2	1	0	0	1	0	2	0	1	0	1	0	1	0	3
27	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	3	0	1
28	0	1	0	2	1	1	0	1	0	1	0	1	0	2	0	1	0	1	0	1	0	1	0	1	0	1	0	5	0	2
29	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	4	1	0	0	1	0	1	0	3	0	1	0	1
30	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	2	0	1	0	1	0	1	0	1	0	1

5.2.6 Predictive Performance Analysis

With the data distribution information provided by the clustering algorithm, we can explain a predictive behavior of each learning algorithm as follows. As *k*-NN uses a majority vote from all neighbors to predict an output, the prediction of *k*-NN is depended on a ratio between the number of susceptible samples and resistant samples near a new sample. Since in most of datasets the density of susceptible samples was higher than resistant samples (as described in section 5.2.6), *k*-NN tends to predict output as a susceptible class.

Although the contribution from each of the hidden nodes of the RBF network is localized to a region nearby the new sample like *k*-NN, the RBF network provides a global approximation to the target function. In addition, in contrast to *k*-NN, the RBF network computes the weights of each hidden nodes (training samples) using all training data in the training step. For this reason, the RBF network can improve the performance of recognizing resistant samples of *k*-NN caused by an imbalance of the density between susceptible samples and resistant samples.

As SVM uses only support vectors (samples) that lie at the border between susceptible and resistant samples and uses an optimization technique to find a suitable hyperplane to classify a new sample, this approach can eliminate a predictive bias of predicting susceptible class which may occur in *k*-NN. Thus SVM provided the best performance in predicting phenotypic HIV-1 drug resistance.

For the CBA algorithm, the predictive behavior does not depend on the distribution of the training data. This is because the CBA algorithm does not use the distance function to produce an output. On the other hand, the prediction of this technique depends on the number of the samples for each class of all training data. Considering the CBA results in Table 5.4 and the percentage of susceptible and resistant classes in Table 3.3 in Chapter 3, we found that for most of drugs, if the number of the susceptible (resistant) sample is higher than the resistant (susceptible) sample in a dataset, the specificity (sensitivity) of that dataset is greater than the sensitivity (specificity).

5.3 Composite Classifier Results and Discussion

5.3.1 Experimental Results

Table 5.7 shows the comparison between the proposed composite classifier (represented by column DCC) and three single classifiers. The experimental results demonstrate that our proposed classifier combination method (DCC) provided the best or the second best accuracy for all drugs. The accuracies of the composite classifier were the best for eleven drugs, the second best for four drugs. In addition, the composite classifier yielded the highest average accuracy.

Table 5.7: The accuracy of three single classifiers and the dynamic composite classifier.

Drug	Accuracy (%)			
	DCC	SVM	RBF network	*k*-NN
LPV	**89.97**	88.40	89.03	87.46
APV	**88.91**	88.17	87.99	84.66
NFV	92.49	**93.13**	92.01	92.17
IDV	93.28	**93.45**	91.93	92.44
SQV	**91.42**	90.76	89.44	87.62
RTV	95.11	**95.46**	94.07	92.15
3TC	**91.68**	91.68	90.55	91.49
ABC	**86.96**	86.58	86.77	83.74
AZT	**93.37**	93.18	90.91	91.67
d4T	**87.17**	86.04	86.60	86.23
ddC	**84.77**	**84.77**	83.25	83.76
ddI	**82.20**	79.17	80.49	80.30
DLV	89.89	**90.07**	88.81	88.09
EFV	**94.85**	94.32	**94.85**	90.59
NVP	**93.59**	92.72	93.07	90.30
average	**90.38**	89.86	89.32	88.18

Table 5.8: The predictive accuracy of single classifiers and the composite classifiers.

Drug	Accuracy (%)					
	SVM	RBF Network	*k*-NN	Majority Vote	Naïve Bayes	DCC
LPV	88.40	89.03	87.46	89.34	89.66	**89.97**
APV	88.17	87.99	84.66	87.43	87.43	**88.91**
NFV	**93.13**	92.01	92.17	**93.13**	**93.13**	92.49
IDV	93.45	91.93	92.44	**93.78**	**93.78**	93.28
SQV	90.76	89.44	87.62	90.92	90.92	**91.42**
RTV	**95.46**	94.07	92.15	**95.46**	95.29	95.11
3TC	**91.68**	90.55	91.49	91.30	91.30	**91.68**
ABC	86.58	86.77	83.74	86.58	86.58	**86.96**
AZT	93.18	90.91	91.67	92.42	92.42	**93.37**
d4T	86.04	86.60	86.23	86.98	**87.36**	87.17
ddC	**84.77**	83.25	83.76	84.52	84.52	**84.77**
ddI	79.17	80.49	80.30	81.44	81.44	**82.20**
DLV	90.07	88.81	88.09	**90.61**	**90.61**	89.89
EFV	94.32	**94.85**	90.59	94.32	94.32	**94.85**
NVP	92.72	93.07	90.30	92.37	93.07	**93.59**
average	89.86	89.32	88.18	90.04	90.12	**90.38**

Table 5.8 demonstrates the predictive accuracy of three single classifiers and three different classifier combination methods. From the results, we found that all composite classifiers enhanced the predictive performance of three single classifiers especially for LPV, SQV, d4T, and ddI drugs in which all of three composite classifiers provided the higher accuracies than three single classifiers. In addition, when comparing the predictive performance of our proposed classifier combination method (DCC) with other two classifier combination methods (Majority Vote and Naive Bayes), we found that DCC provided the best accuracy for ten drugs, and also yielded the highest average accuracy. The accuracy of each data fold for majority vote, naïve Baye, and DCC are shown in Tables A.5-A.7 in Section A.1 of Appendix A.

5.3.2 Predictive Performance Analysis for the Composite Classifier

There are three main reasons that we selected SVM, the RBF network, and *k*-NN to be the component classifiers. First of all, these learning algorithms use the same input data for constructing the models. This prevents a bias of getting different information from input data among three classifiers. The second reason is of the accuracy of the component classifiers. This is an important criterion for selecting component classifiers. The results in Table 5.7 show that all of these classifiers provided the average predictive accuracy greater than 80%. The final reason is of the diversity of the component classifiers. Ali and Pazzani (1996) have shown that error is mostly reduced by using component classifiers whose errors are low correlated. To measure the diversity of SVM, the RBF network, and *k*-NN, we calculated error correlation between all pairs of these learning algorithms as shown in Table 5.10. Since the average of all error correlation between pairs of algorithms was not highly correlated (lower than 0.526), SVM, the RBF network, and *k*-NN were considered to be good candidates for the component classifiers.

To analyze how the composite classifiers enhance the predictive performance of the single classifiers, a static composite classifier was constructed. The static composite classifier combined SVM, the RBF network, and *k*-NN classifiers with weight voting to predict all testing data. Then we applied the Dynamic Weighted Voting (DWV) algorithm already described in Section 4.2.2 to the final prediction of the composite classifier. Table 5.9 shows the accuracy of the static composite classifier compared with three single classifiers. The column *stdev* shows the standard deviations of accuracy of all base classifiers. We used these values to measure the performance variation between the base classifiers.

The results in Tables 5.9 and 5.10 indicate that the improvement of the composite classifier depends on error correlation and performance variation between base classifiers. If the performance variation is small, the improvement could be obtained more easily. Otherwise the base classifier with the worst performance could induce poor performance of the composite classifier. Error correlation is another factor which affects the improvement of the composite classifier. If error correlation is high, the

improvement could not be easily achieved as when one classifier makes error, the others are likely to commit the same error.

In our case, when the standard deviation was small (e.g. less than or equal to 1.0), the improvement was obtained for all drugs except for 3TC and ddC. The predictive performance of the static composite classifiers of 3TC and ddC did not improve because all pairs of classifiers of 3TC and ddC drugs had high error correlation more than 0.7 and 0.8, respectively. On the other hand, when the performance variation was large, it was more difficult to achieve the improvement of the composite classifier, e.g. as in the cases of APV, ABC, AZT, EFV, and NVP. However, when the performance variation was quite large but the error correlation was small, the predictive performance of the composite classifiers could be improved such as in the case of SQV.

Table 5.9: The accuracy of three single classifiers and the static composite classifiers.

Drug	Accuracy (%)				stdev
	SVM RBF network *k*-NN	SVM	RBF network	*k*-NN	
LPV	**89.34**	88.40	89.03	87.46	0.79
APV	87.43	**88.17**	87.99	84.66	1.98
NFV	**93.13**	**93.13**	92.01	92.17	0.60
IDV	**93.78**	93.45	91.93	92.44	0.77
SQV	**90.92**	90.76	89.44	87.62	1.57
RTV	**95.46**	**95.46**	94.07	92.15	1.66
3TC	91.30	**91.68**	90.55	91.49	0.61
ABC	86.58	86.58	**86.77**	83.74	1.69
AZT	92.42	**93.18**	90.91	91.67	1.16
d4T	**86.98**	86.04	86.60	86.23	0.29
ddC	84.52	**84.77**	83.25	83.76	0.78
ddI	**81.44**	79.17	80.49	80.30	0.72
DLV	**90.61**	90.07	88.81	88.09	1.00
EFV	94.14	94.32	**94.85**	90.59	2.32
NVP	92.37	92.72	**93.07**	90.30	1.51
average	90.03	89.86	89.32	88.18	0.86

Table 5.10: Error correlation of all pairs of three algorithms.

Drug	Error correlation			
	SVM RBF network	RBF network *k*-NN	SVM *k*-NN	average
LPV	0.532	0.389	0.510	0.477
APV	0.518	0.410	0.531	0.486
NFV	0.632	0.650	0.704	0.662
IDV	0.403	0.431	0.448	0.427
SQV	0.446	0.418	0.489	0.451
RTV	0.429	0.386	0.392	0.402
3TC	0.741	0.759	0.712	0.737
ABC	0.533	0.444	0.495	0.491
AZT	0.500	0.559	0.509	0.523
d4T	0.480	0.500	0.615	0.532
ddC	0.853	0.831	0.938	0.874
ddI	0.430	0.294	0.518	0.414
DLV	0.539	0.422	0.635	0.532
EFV	0.525	0.323	0.371	0.406
NVP	0.414	0.412	0.581	0.469
average	0.531	0.482	0.563	0.526

To enhance the predictive performance of the composite classifiers, we constructed the dynamic composite classifiers instead of static combination. In our dynamic classifier combination method, a combination pattern of the base classifiers depends on a new instance. This property makes a dynamic composite classifier have more predictive performance than a static composite classifier since the dynamic composite classifier is more adaptable to each new instance.

The important problem of constructing a dynamic composite classifier is how to select the component classifiers. Which base classifiers are suitable to form a composite classifier? To solve this problem, we consider that the accuracy, the performance variation, and the error correlation of the composite classifiers will help us measure the quality of the component classifiers. In addition, the confident weight of each base classifier in predicting a new instance is an important factor in selecting suitable composite classifiers.

Considering the predictive results of the dynamic composite classifiers constructed by DCC (shown in Table 5.7) and the results of the static composite classifiers (shown in Table 5.9), we found that the dynamic composite classifiers enhanced the predictive performance of the static composite classifiers for eleven drugs. Furthermore, our dynamic composite classifiers also yielded the better performance than majority vote and naïve Bayes methods for ten drugs. These results indicate that DCC has an ability to select suitable component classifiers.

CHAPTER VI

CONCLUSIONS

This thesis applies learning algorithms: SVM, the RBF network, *k*-NN, and CBA for constructing the models to predict HIV-1 drug resistance from HIV-1 genotypic data into two classes i.e., resistant or susceptible for 15 drugs separately. The advantage of using the learning algorithm to construct the model is the prediction time. The learning algorithm takes less time than phenotypic testing in prediction. Moreover, the model generated from a learning algorithm helps reduce the cost of phenotypic testing. However, the performance of the learning algorithm depends on the amount of phenotypic training data. The more phenotypic data, the more accuracy of the learning algorithm gains.

For constructing the single classifiers, some pre-processing data techniques such as data selection, data transformation are used to prepare the data suitable for each learning algorithms. In this thesis, RReliefF is applied to select important amino acid positions. From the experimental results, we found that SVM provided the best predictive performance. The method that yielded the second best predictive performance was the RBF network. Moreover, the RBF network had the best ability in recognizing resistant samples. The third best algorithm was CBA. Though *k*-NN had the lowest average accuracy, the predictive performance was quite good (the average accuracy was more than 88.0%). In addition, *k*-NN performed the best performance on recognizing susceptible samples. Besides comparing the predictive performance among four learning algorithms, we also compared the performance of four learning algorithm with online drug resistance systems: HIVdb and Geno2Pheno. The results showed that all learning algorithms provided the better predictive performance than those two online systems.

In the part of the composite classifier construction, this thesis proposes a new dynamic classifier combination method called DCC. The concept of DCC consists of two steps. First, it tries to select the suitable classifiers to form the composite classifier. These classifiers are dynamically chosen by a heuristic function depending on the prediction of each base classifier in classifying a new instance. Then it uses a dynamic weighted voting algorithm to classify the new instance. The results from our experiments indicated that our dynamic composite classifiers enhanced the performance of single classifiers. In addition, the proposed dynamic classifier combination method yielded the better performance than other classifier combination methods such as majority vote and naïve Bayes.

In the HIV treatment, a patient is treated with a combination of drugs. This is an open problem in the application of HIV drug resistance prediction. It is challenging to construct the model to predict the clinical success for drug combination treatments. It would help the doctor finding the combination of drugs that would decrease the viral load of the patients.

REFERENCES

Ali, K.M. and Pazzani, M.J. Error Reduction through Learning Multiple Descriptions. Machine Learning 24 (September 1996): 173-202.

Agrawal, R., Imielinski, T. and Sawami, A. Mining Association Rules between Sets of Items in Large Database. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data 22 (June 1993): 207-216.

Agrawal, R. and Srikant, R. Fast Algorithms for Mining Association Rules. Proceedings of the 20th International Conference on Very Large Data Bases (1994): 487-499.

Battiti, R. and Colla, A.M. Democracy in Neural Nets: Voting Schemes for Classification. Neural Networks 7 (1994): 691 - 707.

Beerenwinkel, N., Daumer, M., Oette, M., Korn, K., Hoffmann, D., Kaiser, R., Lengauer, T., Selbig, J. and Walter, H. Geno2Pheno: Estimating Phenotypic Drug Resistance from HIV-1 Genotypes. Nucleic Acids Research 31 (July 2003): 3850-3855.

Beerenwinkel, N., Schmidt, B., Walter, H., Kaiser, R., Lengauer, T., Hoffmann D., Korn, K. and Selbig, J. Geno2pheno: Interpreting Genotypic HIV Drug Resistance Test. IEEE Intelligent System 16 (November-December 2001): 35-41.

Beerenwinkel, N., Schmidt, B., Walter, H., Kaiser, R., Lengauer, T., Hoffmann, D., Korn, K. and Selbig, J. Diversity and Complexity of HIV-1 Drug Resistance: A Bioinformatics Approach to Prediction Phenotype from Genotype. Proceedings of the National Academy of Sciences (June 2002): 8271-8276.

Boser, B.E., Guyon, I.M. and Vapnik, V. A Training Algorithm for Optimal Margin Classifiers. Fifth Annual Workshop on Computational Learning Theory (1992): 144–152.

Broomhead, D. S. and Lowe, D. Multivariable Functional Interpolation and Adaptive Networks. Complex System 2 (1988): 321-355.

Burges, C. A Tutorial on Support Vector Machine for Pattern Recognition. Data Mining and Knowledge Discovery 2 (June 1998): 121-167.

Coffin, J.M. HIV Population Dynamics in Vivo: Implications for Genetic Variation, Pathogenesis, and Therapy. Science 267 (January 1995): 483-489.

Cortes, C. and Vapnik, V. Support-Vector Networks. Machine Learning 20 (September 1995): 273-297.

Davis, L., Hawkins, J., Maetschke, S. and Bodén, M. Comparing SVM Sequence Kernels: A Protein Subcellular Localization Theme. ACM International Conference Proceeding Series 246 (2006): 39-47.

Demeter, L. and Haubrich, R. Phenotypic and Genotypic Resistance Assays: Methodology, Reliability, and Interpretations. Journal of Acquired Immune Deficiency Syndromes 26 (March 2001): S3-S9.

Dietterich, T. Machine Learning Research: Four Current Directions. AI Magazine 18 (Winter 1997): 97-136.

Draghici, S. and Potter, B. Predicting HIV Drug Resistance with Neural Networks. Bioinformatics 19 (January 2003): 98-107.

Gama, J. Combining Classification Algorithms. PhD's Thesis, University of Porto, 2000.

Ghosh, A. and Parai, B. Protein Secondary Structure Prediction Using Distance Based Classifiers. International Journal of Approximate Reasoning 47 (January 2008): 37-44.

Hansen, L.K. and Salamon, P. Neural Networks Ensembles. Transactions on Pattern Analysis and Machine Intelligence 12 (October 1990): 993-1001.

Haykin, S. Neural Networks: A Comprehensive Foundation. 2nd ed. United States of America: Prentice-Hall, 1999.

Hilario, M., Mitchell, A., Kim, J.H., Bradley, P. and Attwood, T. Classifying Protein Fingerprints. Lecture Notes in Computer Science 3202 (November 2004): 197-208.

Huang, Y., Mccullagh, P., Black, N. and Harper, R. Feature Selection and Classification Model Construction on Type 2 Diabetic Patients' Data. Artificial Intelligence in Medicine 41 (November 2007): 251-262.

Huang, D.S., Zhao, X.M., Huang, G.B. and Cheung, Y.M. Classifying Protein Sequences Using Hydropathy Blocks. Pattern Recognition 39 (December 2006): 2293-2300.

James D. S. and Pharm, D. The Protease Inhibitor Drugs[Online]. (n.d.). Available from: http://www.thebody.com/content/art880.html [2008, February 18]

Jones, D.T. Protein Secondary Structure Prediction Based on Position-Specific Scoring Matrices. Journal of Molecular Biology 292 (September 1999):195-202.

Kira, K. and Rendell, L.A. A Practical Approach to Feature Selection. Proceedings of the ninth International Workshop on Machine Learning (1992): 249-256.

Kononenko, I. Estimating Attributes: Analysis and Extensions of RELIEF. Proceedings of the European Conference on Machine Learning (1994): 171-182.

Kuncheva, L.I. Switching Between Selection and Fusion in Combining Classifiers: An Experiment. IEEE Transactions on Systems, Man, and Cybernetics 32 (April 2002): 146-156.

Kwok, S.W. and Carter, C. Multiple Decision Trees. Proceedings of the Fourth Annual Conference on Uncertainty in Artificial Intelligence (1990): 327–335.

Laethem, K. V., Luca A. K., Antinori, A., et al. A Genotypic Drug Resistance Interpretation Algorithm that Significantly Predicts Therapy Response in HIV-1 Infected Patients. Antiviral Therapy 7 (June 2002): 123-129.

Liu, B., Hsu, W. and Ma, Y. Integrating Classification and Association Rule Mining. Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (1998): 27-31.

Liu, H., Zhu, D. and Feng, H. An Ensemble Classifier for Predicting Eukaryotic Protein Subcellular Locations. The 1st International Conference on Bioinformatics and Biomedical Engineering (July 2007): 168-171.

Luts, J., Heerschap, A., Suykens, A.K.J. and Huffel, S.V. A Combined MRI and MRSI Based Multiclass System for Brain Tumour Recognition Using LS-SVMs with Class Probabilities and Feature Selection. Artificial Intelligence in Medicine 40 (June 2007): 87-102.

Meynard, J. L., Vray, M., Morand, J. L., et al. Phenotypic or Genotypic Resistance Testing for Choosing Antiretroviral Therapy after Treatment Failure: A Randomized Trial. AIDS 16 (March 2002): 727-736.

Mitchell, T.M. Machine Learning. United States of America: McGraw-Hill, 1997.

Moody, J. and Darken, C. J. Fast Learning in Networks of Locally-Tuned Processing Units. Neural Computation 1 (1989): 281-294.

Powell, M. Radial Basis Function for Multivariable Interpolation: A Review. Algorithms for approximation (1987): 143-167.

Quinlan, R. C4.5: Program for Machine Learning. Morgan Kaufmann, 1992.

Reid, C., Bassett, R., Day, S., et al. A Dynamic Rules-Based Interpretation System Derived by an Expert Panel is Predictive of Virological Failure. Antiviral Therapy 7 (2002): s91.

Rhee, S.Y., Gonzales, M.J., Kantor, R., Betts, B.J., Revela, J. and Shafer, R.W. Human Immunodeficiency Virus Reverse Transcriptase and Protease Sequence Database. Nucleic Acids Research 31 (January 2003): 298-303.

Robnik-Sikonja, M. and Kononenko, I. An Adaptation of Relief for Attribute Estimation in Regression. Proceedings of the Fourteenth International Conference on Machine Learning (1997): 296-304.

Seattle Treatment Education Project. Know Your HIV Drugs: NRTIs (The Nukes)[Online]. (n.d.). Available from: http://www.thebody.com/content/art2020.html [2008, February 18]

Seattle Treatment Education Project. Know Your HIV Drugs: NNRTIs[Online]. (n.d.). Available from: http://www.thebody.com/content/art2010.html [2008, February 18].

Sevin, A. D., DeGruttola, V., Nijhuis, M., Schapiro, J. M., Foulkes, A. S., Para, M. F. and Boucher, C. A. B. Methods for Investigation of the Relationship Between Drug-Susceptibility Phenotype and Human Immunodeficiency Virus Type 1 Genotype with Applications to AIDS Clinical Trials Group 333. The Journal of Infectious Diseases 182 (2000): 59-67.

Shafer,R.W., Jung, D.R. and Betts, B.J. Human Immunodeficiency Virus Type 1 Reverse Transcriptase and Protease Mutation Search Engine for Queries. Nature Medicine 6 (2000): 1290-1292.

Shen, H.B. and Chou, K.C. Ensemble Classifier for Protein Fold Pattern Recognition. Bioinformatics 22 (July 2006): 1717-1722.

Skalak, D.B. Prototype Selection for Composite Nearest Neighbor Classifiers. PhD's Thesis, University of Massachusetts, 1997.

Stepenosky, N, Green, D, Kounios, J., Clark, C.M. and Polikar, R. Majority Vote and Decision Template Based Ensemble Classifiers Trained on Event Related Potentials for Early Diagnosis of Alzheimer's Disease. IEEE International Conference on Acoustics, Speech, and Signal Processing (May 2006): 901-904.

Tsymbal, A., Pechenizkiy, M., Cunningham, P. and Puuronen, S. Dynamic Integration of Classifiers for Handling Concept Drift. Special Issue of Information Fusion journal 9 (January 2008): 56-68.

Vapnik, V. Statistical Learning Theory. New York: Wiley, 1998.

Wang, K., Jenwitheesuk, E., Samudrala, R. and Mitter, J.E. Simple Linear Model Provides Highly Accurate Genotypic Predictions of HIV-1 Drug Resistance. Antiviral Therapy 9 (2004): 343-352.

Wang, D. and Larder, B. Enhanced Prediction of Lopinavir Resistance from Genotype by Use of Artificial Neural Networks. Infectious Disease 188 (2003): 653-660.

Wikipedia. Support Vector Machine[Online]. (n.d.). Available from: http://en.wikipedia.org/wiki/Support_vector_machine[2008, April 25]

Wikipedia. The Human Immunodeficiency Virus[Online]. (n.d.). Available from: http://en.wikipedia.org/wiki/HIV [2008, February 18]

Wolpert, D. Stacked Generalization. Neural Network 5 (1992): 241-259.

Yeni, P. G., Hammer, S. M., Carpenter,C. C. J., Cooper,D. A., Fischl, M. A., Gatell, J. M., et al. Antiretroviral Treatment for Adult HIV Infection in 2002 Update Recommendations of the Internation AIDS Society-USA Panel. The Journal of the American Medical Association 288 (July 2002): 222-235.

Zhang, J. Selecting Typical Instances in Instance-Based Learning. Proceedings of the Ninth International Conference on Machine Learning (1992): 470-479.

APPENDICES

APPENDIX A

ADDITIONAL EXPERIMENTAL RESULTS

A.1 The accuracy of 10-Fold Cross-Validation

Tables A.1 to A.7 show the accuracy of 10 folds for the single classifiers and the composite classifiers.

Table A.1: The accuracy of 10 folds for CBA classifiers.

1	90.63	94.56	90.48	98.33	100.00	93.10	84.91	90.57	83.05	94.34	81.82	79.25	81.93	96.49	91.13
2	91.25	95.12	90.48	96.67	91.61	91.38	86.79	86.79	94.34	92.86	87.01	92.59	95.71	96.74	94.83
3	87.01	87.04	90.16	90.01	88.53	93.08	96.34	92.45	91.07	81.13	80.01	69.81	81.07	87.22	87.93
4	90.63	90.13	90.48	93.33	88.53	93.08	86.79	84.91	92.45	83.02	86.01	73.58	86.29	87.01	90.55
5	90.63	88.83	90.48	90.01	87.44	96.46	83.68	88.10	93.11	86.41	86.01	84.91	90.36	91.07	90.31
6	87.01	79.63	90.33	94.01	90.73	97.22	96.34	73.66	93.11	91.07	86.01	75.47	81.93	93.86	94.83
7	87.01	90.74	91.56	89.46	90.67	93.08	93.07	91.13	93.08	86.41	86.01	83.02	83.64	89.23	90.55
8	87.01	88.83	90.16	89.14	90.67	93.08	92.45	84.91	93.07	86.41	77.01	88.03	89.00	89.23	90.40
9	68.75	79.63	91.04	93.33	93.53	93.08	83.68	90.57	93.46	93.07	82.01	88.46	70.36	93.86	92.93
10	77.42	85.03	85.29	92.22	95.03	92.36	93.39	92.69	91.29	93.02	85.01	75.01	82.72	92.35	91.72
average	86.65	87.86	90.08	93.43	90.25	94.04	89.79	85.10	92.04	89.00	83.46	79.13	87.01	91.47	90.65

Table A.2: The accuracy of 10 folds for SVM classifiers.

1	91.25	94.56	97.20	96.67	89.03	93.27	92.45	92.45	90.57	90.57	81.82	79.25	87.50	96.74	93.10
2	94.38	93.09	93.65	96.67	90.15	94.10	93.03	94.91	92.45	92.45	90.63	75.47	94.61	96.74	93.10
3	90.63	93.74	97.49	93.33	88.53	96.65	96.34	96.23	90.57	73.59	87.51	69.81	91.07	96.74	87.93
4	96.25	93.95	92.06	91.07	90.15	96.74	88.03	94.91	92.45	85.69	90.03	79.25	92.86	89.23	93.25
5	93.75	93.31	93.89	93.33	90.15	96.74	96.34	96.23	96.11	85.69	87.51	84.01	100.00	92.55	93.96
6	90.63	93.31	94.83	93.33	90.03	92.99	92.45	79.25	96.11	86.79	87.51	81.13	87.27	96.74	94.83
7	90.63	93.95	95.16	93.33	91.07	93.25	88.03	94.91	92.45	77.36	90.01	81.13	85.46	96.43	94.66
8	81.25	92.06	95.16	96.01	90.01	96.89	96.34	94.91	90.57	86.79	87.51	77.36	92.75	93.01	94.66
9	81.25	93.31	91.94	93.33	90.01	93.25	90.03	90.57	90.57	90.57	87.51	90.56	85.46	92.86	91.21
10	87.10	93.95	93.79	93.33	88.53	92.65	92.11	90.77	90.15	81.13	90.01	71.08	83.74	96.74	93.79
average	89.10	93.34	92.99	93.44	90.15	95.46	91.03	90.57	91.13	86.39	84.70	76.17	93.06	94.11	92.89

Table A.3: The accuracy of 10 folds for RBF network classifiers.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1	91.75	95.56	93.42	96.67	100.00	96.23	92.86	94.74	93.67	96.73	91.82	94.67	97.50	95.45	95.33
2	84.13	87.01	91.48	88.11	86.53	91.13	90.67	84.91	92.45	89.63	90.68	73.09	89.09	92.38	90.81
3	87.50	87.64	90.97	88.11	87.50	89.66	90.67	92.45	89.68	89.10	89.67	77.47	87.50	96.74	90.71
4	100.00	94.94	92.36	96.11	86.50	94.74	88.89	84.91	83.02	89.02	92.60	83.01	91.07	97.60	90.66
5	97.50	93.74	92.36	93.33	95.00	94.74	90.67	97.73	92.45	94.21	95.00	94.34	96.25	96.94	95.66
6	90.73	93.86	95.14	94.90	91.44	91.23	92.67	77.47	88.11	84.91	91.00	79.15	81.64	95.43	90.65
7	90.63	85.19	94.32	87.75	88.33	90.00	88.67	96.23	83.50	91.91	77.50	84.97	91.54	94.21	89.01
8	91.75	89.86	95.97	92.75	85.13	97.40	90.67	91.00	94.32	94.01	92.67	71.79	90.91	96.43	90.65
9	75.00	79.61	93.71	91.61	90.00	92.98	88.89	88.68	83.46	94.34	93.00	76.85	81.64	94.64	89.17
10	87.50	93.31	95.10	95.00	95.00	97.23	92.71	98.64	92.31	93.67	92.50	78.52	94.55	95.42	95.74
average	89.00	87.97	92.91	91.97	89.43	94.07	90.95	90.77	90.91	93.50	93.20	80.40	90.01	94.95	93.06

Table A.4: The accuracy of 10 folds for *k*-NN classifiers.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1	96.88	94.65	93.42	96.11	100.00	96.83	91.96	90.67	90.67	88.68	91.82	77.36	97.50	91.21	91.13
2	91.75	85.19	90.86	90.67	86.53	91.30	90.67	75.79	90.67	86.63	92.67	77.36	91.07	92.30	89.66
3	87.50	87.64	93.09	93.33	93.63	93.10	94.65	94.91	90.50	79.65	92.50	75.47	93.91	92.30	90.21
4	100.00	96.19	90.86	91.67	93.63	90.95	89.68	96.70	90.67	90.67	96.00	83.10	96.71	97.39	96.66
5	97.50	79.63	93.48	93.33	90.16	91.23	90.67	94.91	94.23	86.79	97.50	81.13	96.19	89.29	92.93
6	75.13	79.63	93.97	96.00	91.50	92.40	92.49	73.99	95.11	90.79	92.00	83.10	93.04	96.43	91.38
7	91.25	91.45	91.94	88.14	91.77	93.97	90.67	84.91	92.45	91.00	96.00	83.10	93.74	89.29	92.66
8	97.50	88.59	93.97	94.75	91.77	91.23	94.94	93.02	90.67	84.91	97.00	84.91	90.75	89.29	94.74
9	75.00	77.75	91.94	91.63	96.00	96.74	90.67	94.91	89.68	88.68	90.00	82.69	91.63	92.98	91.01
10	87.50	87.04	95.10	89.63	90.00	91.23	92.31	75.92	90.89	84.91	92.50	75.00	92.73	91.07	89.21
average	89.45	85.64	92.80	92.42	92.60	92.84	91.50	90.74	91.50	86.23	93.90	80.30	93.09	94.50	91.13

Table A.5: The accuracy of 10 folds for majority vote classifiers.

FoldID	LPV	APV	NFV	IDV	SQV	RTV	3TC	ABC	AZT	D4T	DDC	DDI	DLV	EFV	NVP
1	90.00	94.55	91.06	93.33	100.00	93.20	92.45	92.45	90.57	90.57	81.82	81.13	87.50	91.74	91.30
2	84.35	95.15	93.55	93.33	93.16	97.88	90.57	84.91	92.45	92.45	92.86	77.36	94.64	94.74	94.33
3	90.63	90.74	93.43	95.00	95.25	94.33	92.45	92.45	86.83	75.25	82.55	71.70	91.07	92.98	84.40
4	100.00	83.89	92.06	95.00	85.25	96.49	88.65	86.79	90.57	86.79	85.00	75.25	91.07	85.71	93.33
5	90.63	75.19	93.43	93.33	91.30	94.74	90.57	84.91	96.23	96.79	87.50	86.79	100.00	91.64	84.55
6	81.75	83.96	95.32	96.61	95.18	93.98	90.47	77.36	95.11	96.79	87.60	81.00	95.00	96.43	94.33
7	87.50	83.33	93.56	93.14	95.00	100.00	86.65	88.67	90.57	79.25	85.00	86.79	83.64	96.43	96.56
8	84.38	90.74	95.35	97.53	90.00	95.49	94.34	96.79	94.34	92.86	77.50	81.82	92.73	96.43	95.40
9	75.00	79.63	91.94	91.53	91.00	96.49	90.57	90.57	90.59	90.57	81.00	85.45	85.45	94.64	89.47
10	90.00	87.04	95.77	93.33	95.57	92.98	92.21	90.77	92.41	96.60	82.50	76.92	89.09	96.43	87.72
average	89.14	87.42	91.94	93.76	93.02	95.47	91.01	86.97	92.12	86.95	84.45	81.44	90.61	91.12	92.07

Table A.6: The accuracy of 10 folds for naïve Baye classifiers.

FoldID	LPV	APV	NFV	IDV	SQV	RTV	3TC	ABC	AZT	D4T	DDC	DDI	DLV	EFV	NVP
1	90.00	94.55	92.65	93.33	100.00	93.20	92.45	92.45	90.57	90.57	81.82	81.13	87.50	91.74	91.30
2	84.35	85.19	91.65	95.33	93.16	91.38	91.67	84.91	90.47	92.26	92.68	77.36	92.64	92.74	94.33
3	90.63	90.74	90.43	95.00	85.25	94.33	92.45	92.45	88.68	79.25	82.50	71.70	91.07	92.98	90.21
4	100.00	83.39	92.06	95.00	85.25	96.49	93.66	86.79	90.57	96.79	85.00	79.25	92.86	85.71	90.55
5	87.50	85.19	90.43	[illegible]	[illegible]	94.74	90.57	84.91	96.23	96.79	87.50	86.79	100.00	91.64	85.66
6	81.75	83.39	96.14	96.61	95.18	91.23	90.46	77.36	98.11	96.79	87.50	83.02	89.09	96.43	96.55
7	90.63	83.33	93.55	96.74	95.00	100.00	86.68	88.68	90.57	79.25	85.00	86.79	83.64	96.43	94.50
8	81.75	90.74	96.14	91.53	90.91	96.49	94.34	86.79	94.34	93.58	77.50	83.02	92.73	96.43	90.48
9	75.00	79.63	91.94	91.53	90.00	96.49	90.57	90.57	90.57	94.34	82.50	80.40	85.45	94.64	89.47
10	87.50	87.04	90.77	89.23	84.47	92.98	92.31	80.77	92.31	93.58	82.50	76.92	89.09	96.43	94.74
average	89.75	87.42	91.14	93.74	93.00	94.39	91.31	86.67	92.40	87.36	84.45	81.44	90.61	92.91	92.16

Table A.7: The accuracy of 10 folds for DCC classifiers.

A.2 Predictive Performance with Clinical Data

This section reports the predictive performance of our four single classifiers and the proposed composite classifier (DCC) with the TruGene system, an FDA-approved genotypic testing system based on rule-based interpretation rules, by using 97 clinical samples.

Tables A.8 to A.13 show the predictive results of clinical data for TruGene, CBA, SVM, the RBF network, *k*-NN, and DCC models, respectively. All of these tables also show the outcomes of the drug susceptibility from patients. Column 'Sample ID' represents patient ID. Columns Drug1 to Drug4 represent the combination of drugs treated to a patient. Columns 'TruGene1' to 'TruGene4' in Table A.8 represent the predictive results from the TruGene system. The value 1 (-1) in these columns (TruGene1-4) represents resistant (susceptible). The column 'Outcome' of these tables shows the results of the drug susceptibility of the patients. The value -1 represents clinical success. This means that after the doctor treats drugs to the patient, the viral load of that patient becomes lower than 50 copies/ml. On the other hand, the value 1 represents clinical failure. This means that the patient is resistant to the drugs since the viral load is greater than 50 copies/ml.

Table A.8: The predictive results of the clinical data from the TruGene system.

Sample ID	Drug 1	Drug 2	Drug 3	Drug 4	TruGene1	TruGene2	TruGene3	TruGene4	Outcome
14-0028	EFV	IDV	RTV	-	-1	-1	-1	0	-1
14-0033	DDI	AZT	IDV	RTV	1	-1	-1	-1	-1
14-0077	AZT	3TC	-	-	-1	-1	0	0	-1
14-0091	D4T	DDI	RTV	SQV	-1	-1	-1	-1	1
14-0115	AZT	3TC	EFV	-	-1	-1	-1	0	-1
14-0119	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0174	D4T	3TC	NVP	-	-1	-1	1	0	-1
14-0179	AZT	ABC	IDV	RTV	-1	-1	-1	-1	-1
14-0303	D4T	3TC	IDV	RTV	-1	-1	-1	-1	1
14-0320	DDI	AZT	IDV	-	-1	-1	-1	0	-1
14-0328	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0354	D4T	DDI	IDV	RTV	-1	-1	-1	-1	-1
14-0379	AZT	ABC	IDV	RTV	-1	-1	-1	-1	-1
14-0411	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0412	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0425	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0477	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0499	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0502	EFV	IDV	RTV	-	-1	-1	-1	0	-1
14-0529	DDI	AZT	EFV	-	-1	1	-1	0	1
14-0554	DDI	IDV	LPV/R	-	1	-1	-1	0	1
14-0569	AZT	3TC	IDV	RTV	-1	-1	-1	-1	-1
14-0617	DDI	AZT	EFV	LPV/R	-1	-1	-1	-1	-1
14-0704	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0753	AZT	3TC	IDV	RTV	1	1	-1	-1	-1
14-0759	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0760	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0792	D4T	DDI	IDV	RTV	1	-1	-1	-1	-1
14-0801	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0861	DDI	AZT	EFV	-	-1	-1	-1	0	-1
14-1000	DDI	3TC	IDV	RTV	-1	-1	-1	-1	-1
14-1176	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
25-0001	DDI	AZT	-	-	1	-1	0	0	1
25-0003	D4T	DDI	-	-	-1	1	0	0	-1
25-0030	DDI	-	-	-	-1	0	0	0	-1

Table A.8: The predictive results of the clinical data from the TruGene system (cont.).

Sample ID	Drug 1	Drug 2	Drug 3	Drug 4	TruGene1	TruGene2	TruGene3	TruGene4	Outcome
25-0089	DDI	AZT	-	-	-1	-1	0	0	-1
25-0102	DDI	-	-	-	-1	0	0	0	-1
25-0118	DDI	AZT	-	-	-1	-1	0	0	-1
25-0124	DDI	NVP	-	-	-1	-1	0	0	-1
25-0133	AZT	3TC	EFV	-	-1	-1	-1	0	-1
25-0139	AZT	ABC	-	-	-1	-1	0	0	-1
25-0197	3TC	EFV	-	-	-1	-1	0	0	-1
25-0213	DDI	AZT	-	-	-1	-1	0	0	1
25-0415	DDI	AZT	-	-	-1	-1	0	0	-1
25-0419	DDI	AZT	-	-	-1	-1	0	0	-1
25-0442	EFV	-	-	-	-1	0	0	0	-1
25-0452	DDI	AZT	-	-	-1	-1	0	0	-1
25-0495	DDI	AZT	-	-	-1	-1	0	0	1
Gen149	D4T	DDI	3TC	LPV/R	1	1	1	-1	-1
Gen162	D4T	DDI	-	-	-1	-1	0	0	1
Gen171	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen181	D4T	ABC	3TC	LPV/R	-1	-1	-1	-1	-1
Gen221	IDV	RTV	-	-	-1	-1	0	0	-1
Gen228	AZT	3TC	IDV	RTV	1	1	-1	-1	-1
Gen232	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen236	D4T	DDI	IDV	RTV	-1	-1	-1	-1	1
Gen259	3TC	EFV	IDV	RTV	1	-1	-1	-1	-1
Gen270	D4T	DDI	LPV/R	-	1	1	-1	0	-1
Gen298	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen300	AZT	3TC	LPV/R	-	1	1	-1	0	-1
Gen316	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen343	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen351	AZT	3TC	IDV	RTV	-1	-1	-1	-1	-1
Gen353	EFV	IDV	RTV	-	-1	1	1	0	-1
Gen363	AZT	3TC	NFV	-	1	-1	-1	0	-1
Gen364	DDI	3TC	IDV	-	-1	-1	-1	0	-1
Gen365	ABC	3TC	-	-	1	1	0	0	-1
Gen380	AZT	3TC	EFV	-	-1	-1	-1	0	1
Gen395	DDI	AZT	IDV	RTV	-1	-1	1	1	-1
Gen407	EFV	-	-	-	-1	0	0	0	-1

Table A.8: The predictive results of the clinical data from the TruGene system (cont.).

Sample ID	Drug 1	Drug 2	Drug 3	Drug 4	TruGene1	TruGene2	TruGene3	TruGene4	Outcome
Gen414	D4T	3TC	-	-	-1	-1	0	0	-1
Gen422	DDI	3TC	EFV	-	-1	-1	-1	0	1
Gen427	D4T	3TC	NVP	-	-1	-1	-1	0	-1
Gen437	3TC	LPV/R	SQV	-	1	-1	-1	0	-1
Gen439	EFV	LPV/R	-	-	-1	-1	0	0	-1
Gen444	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen448	AZT	3TC	IDV	RTV	-1	-1	-1	-1	-1
Gen460	3TC	EFV	LPV/R	-	1	-1	-1	0	-1
Gen480	AZT	3TC	IDV	RTV	1	-1	-1	-1	-1
Gen503	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen54	D4T	DDI	IDV	-	-1	1	1	0	1
Gen571	DDI	3TC	IDV	RTV	-1	-1	-1	-1	-1
Gen577	ABC	3TC	IDV	RTV	1	-1	-1	-1	-1
Gen61	AZT	ABC	3TC	-	-1	1	1	0	-1
Gen643	DDI	AZT	-	-	-1	-1	0	0	-1
Gen645	EFV	RTV	SQV	-	-1	-1	-1	0	-1
Gen667	DDI	3TC	IDV	RTV	-1	-1	-1	-1	-1
Gen668	D4T	ABC	IDV	RTV	-1	-1	-1	-1	-1
Gen670	LPV/R	SQV	-	-	-1	-1	0	0	-1
Gen671	AZT	3TC	-	-	-1	-1	0	0	-1
Gen703	DDI	3TC	EFV	-	-1	-1	-1	0	-1
Gen75	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen752	AZT	3TC	-	-	-1	-1	0	0	-1
Gen767	DDI	3TC	-	-	-1	-1	0	0	-1
Gen88	EFV	LPV/R	-	-	-1	-1	0	0	-1
Out2925861	LPV/R	SQV	-	-	-1	-1	0	0	-1
Out3601978	LPV/R	SQV	-	-	-1	-1	0	0	-1

Table A.9: The predictive results of the clinical data from the CBA model.

Sample ID	Drug 1	Drug 2	Drug 3	Drug 4	CBA1	CBA2	CBA3	CBA4	Outcome
14-0028	EFV	IDV	RTV	-	-1	-1	-1	0	-1
14-0033	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0077	AZT	3TC	-	-	-1	-1	0	0	-1
14-0091	D4T	DDI	RTV	SQV	-1	-1	-1	-1	1
14-0115	AZT	3TC	EFV	-	-1	-1	-1	0	-1
14-0119	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0174	D4T	3TC	NVP	-	-1	-1	-1	0	-1
14-0179	AZT	ABC	IDV	RTV	-1	-1	-1	-1	-1
14-0303	D4T	3TC	IDV	RTV	-1	-1	-1	-1	1
14-0320	DDI	AZT	IDV	-	-1	-1	-1	0	-1
14-0328	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0354	D4T	DDI	IDV	RTV	1	1	-1	-1	-1
14-0379	AZT	ABC	IDV	RTV	-1	-1	-1	-1	-1
14-0411	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0412	DDI	AZT	IDV	RTV	-1	-1	1	-1	-1
14-0425	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0477	DDI	AZT	IDV	RTV	1	-1	-1	-1	-1
14-0499	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0502	EFV	IDV	RTV	-	-1	-1	-1	0	-1
14-0529	DDI	AZT	EFV	-	-1	-1	-1	0	1
14-0554	DDI	IDV	LPV/R	-	-1	-1	-1	0	1
14-0569	AZT	3TC	IDV	RTV	-1	1	-1	-1	-1
14-0617	DDI	AZT	EFV	LPV/R	-1	-1	-1	-1	-1
14-0704	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0753	AZT	3TC	IDV	RTV	1	1	-1	-1	-1
14-0759	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0760	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0792	D4T	DDI	IDV	RTV	-1	-1	-1	-1	-1
14-0801	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0861	DDI	AZT	EFV	-	-1	-1	-1	0	-1
14-1000	DDI	3TC	IDV	RTV	-1	1	-1	-1	-1
14-1176	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
25-0001	DDI	AZT	-	-	-1	-1	0	0	1
25-0003	D4T	DDI	-	-	-1	-1	0	0	-1
25-0030	DDI	-	-	-	-1	0	0	0	-1

Table A.9: The predictive results of the clinical data from the CBA model (cont.).

Sample ID	Drug 1	Drug 2	Drug 3	Drug 4	CBA1	CBA2	CBA3	CBA4	Outcome
25-0089	DDI	AZT	-	-	-1	-1	0	0	-1
25-0102	DDI	-	-	-	-1	0	0	0	-1
25-0118	DDI	AZT	-	-	-1	-1	0	0	-1
25-0124	DDI	NVP	-	-	1	1	0	0	-1
25-0133	AZT	3TC	EFV	-	-1	-1	-1	0	-1
25-0139	AZT	ABC	-	-	-1	-1	0	0	-1
25-0197	3TC	EFV	-	-	-1	-1	0	0	-1
25-0213	DDI	AZT	-	-	-1	-1	0	0	1
25-0415	DDI	AZT	-	-	-1	-1	0	0	-1
25-0419	DDI	AZT	-	-	-1	-1	0	0	-1
25-0442	EFV	-	-	-	-1	0	0	0	-1
25-0452	DDI	AZT	-	-	-1	-1	0	0	-1
25-0495	DDI	AZT	-	-	-1	-1	0	0	1
Gen149	D4T	DDI	3TC	LPV/R	1	1	1	-1	-1
Gen162	D4T	DDI	-	-	-1	-1	0	0	1
Gen171	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen181	D4T	ABC	3TC	LPV/R	-1	-1	-1	-1	-1
Gen221	IDV	RTV	-	-	-1	-1	0	0	-1
Gen228	AZT	3TC	IDV	RTV	-1	-1	-1	-1	-1
Gen232	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen236	D4T	DDI	IDV	RTV	-1	-1	-1	-1	1
Gen259	3TC	EFV	IDV	RTV	1	-1	-1	-1	-1
Gen270	D4T	DDI	LPV/R	-	1	-1	-1	0	-1
Gen298	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen300	AZT	3TC	LPV/R	-	1	1	-1	0	-1
Gen316	EFV	IDV	RTV	-	-1	1	-1	0	-1
Gen343	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen351	AZT	3TC	IDV	RTV	-1	1	-1	-1	-1
Gen353	EFV	IDV	RTV	-	-1	1	1	0	-1
Gen363	AZT	3TC	NFV	-	1	-1	-1	0	-1
Gen364	DDI	3TC	IDV	-	-1	-1	-1	0	-1
Gen365	ABC	3TC	-	-	1	-1	0	0	-1
Gen380	AZT	3TC	EFV	-	-1	-1	-1	0	1
Gen395	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
Gen407	EFV	-	-	-	-1	0	0	0	-1

Table A.9: The predictive results of the clinical data from the CBA model (cont.).

Sample ID	Drug 1	Drug 2	Drug 3	Drug 4	CBA1	CBA2	CBA3	CBA4	Outcome
Gen414	D4T	3TC	-	-	-1	-1	0	0	-1
Gen422	DDI	3TC	EFV	-	1	-1	-1	0	1
Gen427	D4T	3TC	NVP	-	-1	-1	-1	0	-1
Gen437	3TC	LPV/R	SQV	-	-1	-1	-1	0	-1
Gen439	EFV	LPV/R	-	-	-1	-1	0	0	-1
Gen444	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen448	AZT	3TC	IDV	RTV	-1	-1	-1	-1	-1
Gen460	3TC	EFV	LPV/R	-	1	-1	-1	0	-1
Gen480	AZT	3TC	IDV	RTV	1	-1	-1	-1	-1
Gen503	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen54	D4T	DDI	IDV	-	-1	-1	-1	0	1
Gen571	DDI	3TC	IDV	RTV	-1	1	-1	-1	-1
Gen577	ABC	3TC	IDV	RTV	-1	1	-1	-1	-1
Gen61	AZT	ABC	3TC	-	-1	-1	1	0	-1
Gen643	DDI	AZT	-	-	-1	-1	0	0	-1
Gen645	EFV	RTV	SQV	-	-1	-1	-1	0	-1
Gen667	DDI	3TC	IDV	RTV	-1	-1	-1	-1	-1
Gen668	D4T	ABC	IDV	RTV	-1	-1	-1	-1	-1
Gen670	LPV/R	SQV	-	-	-1	-1	0	0	-1
Gen671	AZT	3TC	-	-	-1	1	0	0	-1
Gen703	DDI	3TC	EFV	-	-1	-1	-1	0	-1
Gen75	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen752	AZT	3TC	-	-	-1	-1	0	0	-1
Gen767	DDI	3TC	-	-	-1	-1	0	0	-1
Gen88	EFV	LPV/R	-	-	-1	-1	0	0	-1
Out2925861	LPV/R	SQV	-	-	-1	-1	0	0	-1
Out3601978	LPV/R	SQV	-	-	-1	-1	0	0	-1

Table A.10: The predictive results of the clinical data from the SVM model.

Sample ID	Drug 1	Drug 2	Drug 3	Drug 4	SVM1	SVM2	SVM3	SVM4	Outcome
14-0028	EFV	IDV	RTV	-	-1	-1	-1	0	-1
14-0033	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0077	AZT	3TC	-	-	-1	-1	0	0	-1
14-0091	D4T	DDI	RTV	SQV	-1	-1	-1	-1	1
14-0115	AZT	3TC	EFV	-	-1	-1	-1	0	-1
14-0119	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0174	D4T	3TC	NVP	-	-1	-1	1	0	-1
14-0179	AZT	ABC	IDV	RTV	-1	-1	-1	-1	-1
14-0303	D4T	3TC	IDV	RTV	-1	-1	-1	-1	1
14-0320	DDI	AZT	IDV	-	-1	-1	-1	0	-1
14-0328	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0354	D4T	DDI	IDV	RTV	-1	-1	-1	-1	-1
14-0379	AZT	ABC	IDV	RTV	-1	-1	-1	-1	-1
14-0411	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0412	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0425	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0477	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0499	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0502	EFV	IDV	RTV	-	-1	-1	-1	0	-1
14-0529	DDI	AZT	EFV	-	-1	1	-1	0	1
14-0554	DDI	IDV	LPV/R	-	-1	-1	-1	0	1
14-0569	AZT	3TC	IDV	RTV	-1	-1	-1	-1	-1
14-0617	DDI	AZT	EFV	LPV/R	-1	-1	1	-1	-1
14-0704	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0753	AZT	3TC	IDV	RTV	1	1	-1	-1	-1
14-0759	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0760	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0792	D4T	DDI	IDV	RTV	-1	-1	-1	-1	-1
14-0801	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0861	DDI	AZT	EFV	-	-1	-1	-1	0	-1
14-1000	DDI	3TC	IDV	RTV	1	1	-1	-1	-1
14-1176	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
25-0001	DDI	AZT	-	-	-1	-1	0	0	1
25-0003	D4T	DDI	-	-	-1	1	0	0	-1
25-0030	DDI	-	-	-	-1	0	0	0	-1

Table A.10: The predictive results of the clinical data from the SVM model (cont.).

Sample ID	Drug 1	Drug 2	Drug 3	Drug 4	SVM1	SVM2	SVM3	SVM4	Outcome
25-0089	DDI	AZT	-	-	-1	-1	0	0	-1
25-0102	DDI	-	-	-	-1	0	0	0	-1
25-0118	DDI	AZT	-	-	-1	-1	0	0	-1
25-0124	DDI	NVP	-	-	-1	1	0	0	-1
25-0133	AZT	3TC	EFV	-	-1	-1	-1	0	-1
25-0139	AZT	ABC	-	-	-1	-1	0	0	-1
25-0197	3TC	EFV	-	-	-1	-1	0	0	-1
25-0213	DDI	AZT	-	-	-1	-1	0	0	1
25-0415	DDI	AZT	-	-	-1	-1	0	0	-1
25-0419	DDI	AZT	-	-	1	1	0	0	-1
25-0442	EFV	-	-	-	-1	0	0	0	-1
25-0452	DDI	AZT	-	-	-1	-1	0	0	-1
25-0495	DDI	AZT	-	-	-1	-1	0	0	1
Gen149	D4T	DDI	3TC	LPV/R	1	1	1	-1	-1
Gen162	D4T	DDI	-	-	-1	-1	0	0	1
Gen171	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen181	D4T	ABC	3TC	LPV/R	-1	-1	-1	-1	-1
Gen221	IDV	RTV	-	-	-1	-1	0	0	-1
Gen228	AZT	3TC	IDV	RTV	-1	1	-1	-1	-1
Gen232	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen236	D4T	DDI	IDV	RTV	-1	-1	-1	-1	1
Gen259	3TC	EFV	IDV	RTV	1	-1	-1	-1	-1
Gen270	D4T	DDI	LPV/R	-	1	-1	-1	0	-1
Gen298	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen300	AZT	3TC	LPV/R	-	1	1	-1	0	-1
Gen316	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen343	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen351	AZT	3TC	IDV	RTV	-1	1	-1	-1	-1
Gen353	EFV	IDV	RTV	-	-1	1	-1	0	-1
Gen363	AZT	3TC	NFV	-	1	1	-1	0	-1
Gen364	DDI	3TC	IDV	-	-1	1	-1	0	-1
Gen365	ABC	3TC	-	-	1	-1	0	0	-1
Gen380	AZT	3TC	EFV	-	-1	-1	-1	0	1
Gen395	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
Gen407	EFV	-	-	-	-1	0	0	0	-1

Table A.10: The predictive results of the clinical data from the SVM model (cont.).

Sample ID	Drug 1	Drug 2	Drug 3	Drug 4	SVM1	SVM2	SVM3	SVM4	Outcome
Gen414	D4T	3TC	-	-	-1	-1	0	0	-1
Gen422	DDI	3TC	EFV	-	-1	1	-1	0	1
Gen427	D4T	3TC	NVP	-	-1	-1	-1	0	-1
Gen437	3TC	LPV/R	SQV	-	-1	-1	-1	0	-1
Gen439	EFV	LPV/R	-	-	-1	-1	0	0	-1
Gen444	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen448	AZT	3TC	IDV	RTV	-1	-1	-1	-1	-1
Gen460	3TC	EFV	LPV/R	-	1	-1	-1	0	-1
Gen480	AZT	3TC	IDV	RTV	1	1	-1	-1	-1
Gen503	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen54	D4T	DDI	IDV	-	-1	-1	-1	0	1
Gen571	DDI	3TC	IDV	RTV	-1	1	-1	-1	-1
Gen577	ABC	3TC	IDV	RTV	-1	-1	-1	-1	-1
Gen61	AZT	ABC	3TC	-	-1	-1	1	0	-1
Gen643	DDI	AZT	-	-	-1	-1	0	0	-1
Gen645	EFV	RTV	SQV	-	-1	-1	-1	0	-1
Gen667	DDI	3TC	IDV	RTV	-1	1	-1	-1	-1
Gen668	D4T	ABC	IDV	RTV	-1	-1	-1	-1	-1
Gen670	LPV/R	SQV	-	-	-1	-1	0	0	-1
Gen671	AZT	3TC	-	-	-1	1	0	0	-1
Gen703	DDI	3TC	EFV	-	-1	-1	-1	0	-1
Gen75	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen752	AZT	3TC	-	-	-1	-1	0	0	-1
Gen767	DDI	3TC	-	-	-1	-1	0	0	-1
Gen88	EFV	LPV/R	-	-	-1	-1	0	0	-1
Out2925861	LPV/R	SQV	-	-	-1	-1	0	0	-1
Out3601978	LPV/R	SQV	-	-	-1	-1	0	0	-1

Table A.11: The predictive results of the clinical data from the RBF network model.

Sample ID	Drug 1	Drug 2	Drug 3	Drug 4	RBF1	RBF2	RBF3	RBF4	Outcome
14-0028	EFV	IDV	RTV	-	-1	-1	-1	0	-1
14-0033	DDI	AZT	IDV	RTV	1	-1	-1	-1	-1
14-0077	AZT	3TC	-	-	-1	-1	0	0	-1
14-0091	D4T	DDI	RTV	SQV	-1	-1	-1	-1	1
14-0115	AZT	3TC	EFV	-	-1	-1	-1	0	-1
14-0119	DDI	AZT	IDV	RTV	1	-1	-1	-1	-1
14-0174	D4T	3TC	NVP	-	-1	-1	1	0	-1
14-0179	AZT	ABC	IDV	RTV	-1	-1	-1	-1	-1
14-0303	D4T	3TC	IDV	RTV	-1	-1	-1	-1	1
14-0320	DDI	AZT	IDV	-	1	-1	-1	0	-1
14-0328	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0354	D4T	DDI	IDV	RTV	-1	-1	-1	-1	-1
14-0379	AZT	ABC	IDV	RTV	-1	1	-1	-1	-1
14-0411	DDI	AZT	IDV	RTV	1	-1	-1	-1	-1
14-0412	DDI	AZT	IDV	RTV	1	-1	-1	-1	-1
14-0425	DDI	AZT	IDV	RTV	1	-1	-1	-1	-1
14-0477	DDI	AZT	IDV	RTV	1	-1	-1	-1	-1
14-0499	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0502	EFV	IDV	RTV	-	-1	-1	-1	0	-1
14-0529	DDI	AZT	EFV	-	1	1	1	0	1
14-0554	DDI	IDV	LPV/R	-	1	1	-1	0	1
14-0569	AZT	3TC	IDV	RTV	-1	1	-1	-1	-1
14-0617	DDI	AZT	EFV	LPV/R	1	-1	1	-1	-1
14-0704	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0753	AZT	3TC	IDV	RTV	1	1	-1	-1	-1
14-0759	DDI	AZT	IDV	RTV	1	-1	-1	-1	-1
14-0760	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0792	D4T	DDI	IDV	RTV	-1	1	-1	-1	-1
14-0801	DDI	AZT	IDV	RTV	1	-1	-1	-1	-1
14-0861	DDI	AZT	EFV	-	1	-1	-1	0	-1
14-1000	DDI	3TC	IDV	RTV	1	1	-1	-1	-1
14-1176	DDI	AZT	IDV	RTV	1	-1	-1	-1	-1
25-0001	DDI	AZT	-	-	1	-1	0	0	1
25-0003	D4T	DDI	-	-	-1	-1	0	0	-1
25-0030	DDI	-	-	-	1	0	0	0	-1

Table A.11: The predictive results of the clinical data from the RBF network model (cont.).

Sample ID	Drug 1	Drug 2	Drug 3	Drug 4	RBF1	RBF2	RBF3	RBF4	Outcome
25-0089	DDI	AZT	-	-	1	-1	0	0	-1
25-0102	DDI	-	-	-	1	0	0	0	-1
25-0118	DDI	AZT	-	-	1	-1	0	0	-1
25-0124	DDI	NVP	-	-	-1	1	0	0	-1
25-0133	AZT	3TC	EFV	-	-1	-1	1	0	-1
25-0139	AZT	ABC	-	-	-1	1	0	0	-1
25-0197	3TC	EFV	-	-	-1	-1	0	0	-1
25-0213	DDI	AZT	-	-	1	-1	0	0	1
25-0415	DDI	AZT	-	-	1	-1	0	0	-1
25-0419	DDI	AZT	-	-	1	-1	0	0	-1
25-0442	EFV	-	-	-	-1	0	0	0	-1
25-0452	DDI	AZT	-	-	1	-1	0	0	-1
25-0495	DDI	AZT	-	-	1	-1	0	0	1
Gen149	D4T	DDI	3TC	LPV/R	1	1	1	-1	-1
Gen162	D4T	DDI	-	-	-1	-1	0	0	1
Gen171	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen181	D4T	ABC	3TC	LPV/R	-1	-1	-1	-1	-1
Gen221	IDV	RTV	-	-	-1	-1	0	0	-1
Gen228	AZT	3TC	IDV	RTV	-1	1	-1	-1	-1
Gen232	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen236	D4T	DDI	IDV	RTV	-1	1	-1	-1	1
Gen259	3TC	EFV	IDV	RTV	1	1	-1	-1	-1
Gen270	D4T	DDI	LPV/R	-	1	-1	-1	0	-1
Gen298	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen300	AZT	3TC	LPV/R	-	1	1	-1	0	-1
Gen316	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen343	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen351	AZT	3TC	IDV	RTV	-1	1	-1	-1	-1
Gen353	EFV	IDV	RTV	-	-1	1	-1	0	-1
Gen363	AZT	3TC	NFV	-	1	1	-1	0	-1
Gen364	DDI	3TC	IDV	-	1	1	-1	0	-1
Gen365	ABC	3TC	-	-	1	-1	0	0	-1
Gen380	AZT	3TC	EFV	-	-1	-1	-1	0	1
Gen395	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
Gen407	EFV	-	-	-	1	0	0	0	-1

Table A.11: The predictive results of the clinical data from the RBF network model (cont.).

Sample ID	Drug 1	Drug 2	Drug 3	Drug 4	RBF1	RBF2	RBF3	RBF4	Outcome
Gen414	D4T	3TC	-	-	-1	-1	0	0	-1
Gen422	DDI	3TC	EFV	-	1	1	-1	0	1
Gen427	D4T	3TC	NVP	-	-1	-1	-1	0	-1
Gen437	3TC	LPV/R	SQV	-	-1	-1	-1	0	-1
Gen439	EFV	LPV/R	-	-	-1	-1	0	0	-1
Gen444	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen448	AZT	3TC	IDV	RTV	-1	-1	-1	-1	-1
Gen460	3TC	EFV	LPV/R	-	1	-1	-1	0	-1
Gen480	AZT	3TC	IDV	RTV	1	1	-1	-1	-1
Gen503	EFV	IDV	RTV	-	-1	1	-1	0	-1
Gen54	D4T	DDI	IDV	-	-1	1	1	0	1
Gen571	DDI	3TC	IDV	RTV	1	1	-1	-1	-1
Gen577	ABC	3TC	IDV	RTV	-1	-1	-1	-1	-1
Gen61	AZT	ABC	3TC	-	-1	-1	1	0	-1
Gen643	DDI	AZT	-	-	1	-1	0	0	-1
Gen645	EFV	RTV	SQV	-	-1	-1	-1	0	-1
Gen667	DDI	3TC	IDV	RTV	-1	-1	-1	-1	-1
Gen668	D4T	ABC	IDV	RTV	-1	-1	-1	-1	-1
Gen670	LPV/R	SQV	-	-	-1	-1	0	0	-1
Gen671	AZT	3TC	-	-	-1	1	0	0	-1
Gen703	DDI	3TC	EFV	-	-1	-1	-1	0	-1
Gen75	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen752	AZT	3TC	-	-	-1	-1	0	0	-1
Gen767	DDI	3TC	-	-	1	-1	0	0	-1
Gen88	EFV	LPV/R	-	-	-1	-1	0	0	-1
Out2925861	LPV/R	SQV	-	-	-1	-1	0	0	-1
Out3601978	LPV/R	SQV	-	-	-1	-1	0	0	-1

Table A.12: The predictive results of the clinical data from the *k*-NN model.

Sample ID	Drug 1	Drug 2	Drug 3	Drug 4	*k*-NN1	*k*-NN2	*k*-NN3	*k*-NN4	Outcome
14-0028	EFV	IDV	RTV	-	-1	-1	-1	0	-1
14-0033	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0077	AZT	3TC	-	-	-1	-1	0	0	-1
14-0091	D4T	DDI	RTV	SQV	-1	-1	-1	-1	1
14-0115	AZT	3TC	EFV	-	-1	-1	-1	0	-1
14-0119	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0174	D4T	3TC	NVP	-	-1	-1	1	0	-1
14-0179	AZT	ABC	IDV	RTV	-1	-1	-1	-1	-1
14-0303	D4T	3TC	IDV	RTV	-1	-1	-1	-1	1
14-0320	DDI	AZT	IDV	-	-1	-1	-1	0	-1
14-0328	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0354	D4T	DDI	IDV	RTV	-1	-1	-1	-1	-1
14-0379	AZT	ABC	IDV	RTV	-1	-1	-1	-1	-1
14-0411	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0412	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0425	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0477	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0499	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0502	EFV	IDV	RTV	-	-1	-1	-1	0	-1
14-0529	DDI	AZT	EFV	-	-1	1	-1	0	1
14-0554	DDI	IDV	LPV/R	-	-1	-1	-1	0	1
14-0569	AZT	3TC	IDV	RTV	-1	-1	-1	-1	-1
14-0617	DDI	AZT	EFV	LPV/R	-1	-1	1	-1	-1
14-0704	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0753	AZT	3TC	IDV	RTV	1	1	-1	-1	-1
14-0759	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0760	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0792	D4T	DDI	IDV	RTV	-1	-1	-1	-1	-1
14-0801	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0861	DDI	AZT	EFV	-	-1	-1	-1	0	-1
14-1000	DDI	3TC	IDV	RTV	-1	1	-1	-1	-1
14-1176	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
25-0001	DDI	AZT	-	-	-1	-1	0	0	1
25-0003	D4T	DDI	-	-	-1	-1	0	0	-1
25-0030	DDI	-	-	-	-1	0	0	0	-1

Table A.12: The predictive results of the clinical data from the *k*-NN model (cont.).

Sample ID	Drug 1	Drug 2	Drug 3	Drug 4	*k*-NN1	*k*-NN2	*k*-NN3	*k*-NN4	Outcome
25-0089	DDI	AZT	-	-	-1	-1	0	0	-1
25-0102	DDI	-	-	-	-1	0	0	0	-1
25-0118	DDI	AZT	-	-	-1	-1	0	0	-1
25-0124	DDI	NVP	-	-	-1	-1	0	0	-1
25-0133	AZT	3TC	EFV	-	-1	-1	-1	0	-1
25-0139	AZT	ABC	-	-	-1	-1	0	0	-1
25-0197	3TC	EFV	-	-	-1	-1	0	0	-1
25-0213	DDI	AZT	-	-	-1	-1	0	0	1
25-0415	DDI	AZT	-	-	-1	-1	0	0	-1
25-0419	DDI	AZT	-	-	-1	-1	0	0	-1
25-0442	EFV	-	-	-	-1	0	0	0	-1
25-0452	DDI	AZT	-	-	-1	-1	0	0	-1
25-0495	DDI	AZT	-	-	-1	-1	0	0	1
Gen149	D4T	DDI	3TC	LPV/R	1	1	1	-1	-1
Gen162	D4T	DDI	-	-	-1	-1	0	0	1
Gen171	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen181	D4T	ABC	3TC	LPV/R	-1	-1	-1	-1	-1
Gen221	IDV	RTV	-	-	-1	-1	0	0	-1
Gen228	AZT	3TC	IDV	RTV	1	1	-1	-1	-1
Gen232	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen236	D4T	DDI	IDV	RTV	-1	-1	-1	-1	1
Gen259	3TC	EFV	IDV	RTV	1	-1	-1	-1	-1
Gen270	D4T	DDI	LPV/R	-	1	1	-1	0	-1
Gen298	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen300	AZT	3TC	LPV/R	-	1	1	-1	0	-1
Gen316	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen343	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen351	AZT	3TC	IDV	RTV	-1	1	-1	-1	-1
Gen353	EFV	IDV	RTV	-	-1	1	-1	0	-1
Gen363	AZT	3TC	NFV	-	1	1	-1	0	-1
Gen364	DDI	3TC	IDV	-	-1	1	-1	0	-1
Gen365	ABC	3TC	-	-	1	-1	0	0	-1
Gen380	AZT	3TC	EFV	-	-1	-1	-1	0	1
Gen395	DDI	AZT	IDV	RTV	-1	-1	1	-1	-1
Gen407	EFV	-	-	-	-1	0	0	0	-1

Table A.12: The predictive results of the clinical data from the *k*-NN model (cont.).

Sample ID	Drug 1	Drug 2	Drug 3	Drug 4	*k*-NN1	*k*-NN2	*k*-NN3	*k*-NN4	Outcome
Gen414	D4T	3TC	-	-	-1	-1	0	0	-1
Gen422	DDI	3TC	EFV	-	-1	1	-1	0	1
Gen427	D4T	3TC	NVP	-	-1	-1	-1	0	-1
Gen437	3TC	LPV/R	SQV	-	-1	-1	-1	0	-1
Gen439	EFV	LPV/R	-	-	-1	-1	0	0	-1
Gen444	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen448	AZT	3TC	IDV	RTV	-1	-1	-1	-1	-1
Gen460	3TC	EFV	LPV/R	-	1	-1	-1	0	-1
Gen480	AZT	3TC	IDV	RTV	1	1	-1	-1	-1
Gen503	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen54	D4T	DDI	IDV	-	-1	-1	-1	0	1
Gen571	DDI	3TC	IDV	RTV	-1	1	-1	-1	-1
Gen577	ABC	3TC	IDV	RTV	-1	-1	-1	-1	-1
Gen61	AZT	ABC	3TC	-	-1	-1	1	0	-1
Gen643	DDI	AZT	-	-	-1	-1	0	0	-1
Gen645	EFV	RTV	SQV	-	-1	-1	-1	0	-1
Gen667	DDI	3TC	IDV	RTV	-1	1	-1	-1	-1
Gen668	D4T	ABC	IDV	RTV	-1	-1	-1	-1	-1
Gen670	LPV/R	SQV	-	-	-1	-1	0	0	-1
Gen671	AZT	3TC	-	-	-1	1	0	0	-1
Gen703	DDI	3TC	EFV	-	-1	-1	-1	0	-1
Gen75	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen752	AZT	3TC	-	-	-1	-1	0	0	-1
Gen767	DDI	3TC	-	-	-1	-1	0	0	-1
Gen88	EFV	LPV/R	-	-	-1	-1	0	0	-1
Out2925861	LPV/R	SQV	-	-	-1	-1	0	0	-1
Out3601978	LPV/R	SQV	-	-	-1	-1	0	0	-1

Table A.13: The predictive results of the clinical data from the DCC model.

Sample ID	Drug 1	Drug 2	Drug 3	Drug 4	DCC1	DCC2	DCC3	DCC4	Outcome
14-0028	EFV	IDV	RTV	-	-1	-1	-1	0	-1
14-0033	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0077	AZT	3TC	-	-	-1	-1	0	0	-1
14-0091	D4T	DDI	RTV	SQV	-1	-1	-1	-1	1
14-0115	AZT	3TC	EFV	-	-1	-1	-1	0	-1
14-0119	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0174	D4T	3TC	NVP	-	-1	-1	1	0	-1
14-0179	AZT	ABC	IDV	RTV	-1	-1	-1	-1	-1
14-0303	D4T	3TC	IDV	RTV	-1	-1	-1	-1	1
14-0320	DDI	AZT	IDV	-	-1	-1	-1	0	-1
14-0328	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0354	D4T	DDI	IDV	RTV	-1	-1	-1	-1	-1
14-0379	AZT	ABC	IDV	RTV	-1	-1	-1	-1	-1
14-0411	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0412	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0425	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0477	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0499	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0502	EFV	IDV	RTV	-	-1	-1	-1	0	-1
14-0529	DDI	AZT	EFV	-	-1	1	-1	0	1
14-0554	DDI	IDV	LPV/R	-	-1	1	-1	0	1
14-0569	AZT	3TC	IDV	RTV	-1	1	-1	-1	-1
14-0617	DDI	AZT	EFV	LPV/R	-1	-1	1	-1	-1
14-0704	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0753	AZT	3TC	IDV	RTV	1	1	-1	-1	-1
14-0759	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0760	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0792	D4T	DDI	IDV	RTV	-1	-1	-1	-1	-1
14-0801	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
14-0861	DDI	AZT	EFV	-	-1	-1	-1	0	-1
14-1000	DDI	3TC	IDV	RTV	1	1	-1	-1	-1
14-1176	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
25-0001	DDI	AZT	-	-	-1	-1	0	0	1
25-0003	D4T	DDI	-	-	-1	-1	0	0	-1
25-0030	DDI	-	-	-	-1	0	0	0	-1

Table A.13: The predictive results of the clinical data from the DCC model (cont.).

Sample ID	Drug 1	Drug 2	Drug 3	Drug 4	DCC1	DCC2	DCC3	DCC4	Outcome
25-0089	DDI	AZT	-	-	-1	-1	0	0	-1
25-0102	DDI	-	-	-	-1	0	0	0	-1
25-0118	DDI	AZT	-	-	-1	-1	0	0	-1
25-0124	DDI	NVP	-	-	-1	-1	0	0	-1
25-0133	AZT	3TC	EFV	-	-1	-1	-1	0	-1
25-0139	AZT	ABC	-	-	-1	-1	0	0	-1
25-0197	3TC	EFV	-	-	-1	-1	0	0	-1
25-0213	DDI	AZT	-	-	-1	-1	0	0	1
25-0415	DDI	AZT	-	-	-1	-1	0	0	-1
25-0419	DDI	AZT	-	-	1	-1	0	0	-1
25-0442	EFV	-	-	-	-1	0	0	0	-1
25-0452	DDI	AZT	-	-	-1	-1	0	0	-1
25-0495	DDI	AZT	-	-	-1	-1	0	0	1
Gen149	D4T	DDI	3TC	LPV/R	1	1	1	-1	-1
Gen162	D4T	DDI	-	-	-1	-1	0	0	1
Gen171	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen181	D4T	ABC	3TC	LPV/R	-1	-1	-1	-1	-1
Gen221	IDV	RTV	-	-	-1	-1	0	0	-1
Gen228	AZT	3TC	IDV	RTV	-1	1	-1	-1	-1
Gen232	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen236	D4T	DDI	IDV	RTV	-1	-1	-1	-1	1
Gen259	3TC	EFV	IDV	RTV	1	-1	-1	-1	-1
Gen270	D4T	DDI	LPV/R	-	1	-1	-1	0	-1
Gen298	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen300	AZT	3TC	LPV/R	-	1	1	-1	0	-1
Gen316	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen343	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen351	AZT	3TC	IDV	RTV	-1	1	-1	-1	-1
Gen353	EFV	IDV	RTV	-	-1	1	-1	0	-1
Gen363	AZT	3TC	NFV	-	1	1	-1	0	-1
Gen364	DDI	3TC	IDV	-	-1	1	-1	0	-1
Gen365	ABC	3TC	-	-	1	-1	0	0	-1
Gen380	AZT	3TC	EFV	-	-1	-1	-1	0	1
Gen395	DDI	AZT	IDV	RTV	-1	-1	-1	-1	-1
Gen407	EFV	-	-	-	-1	0	0	0	-1

Table A.13: The predictive results of the clinical data from the DCC model (cont.).

Sample ID	Drug 1	Drug 2	Drug 3	Drug 4	DCC1	DCC2	DCC3	DCC4	Outcome
Gen414	D4T	3TC	-	-	-1	-1	0	0	-1
Gen422	DDI	3TC	EFV	-	-1	1	-1	0	1
Gen427	D4T	3TC	NVP	-	-1	-1	-1	0	-1
Gen437	3TC	LPV/R	SQV	-	-1	-1	-1	0	-1
Gen439	EFV	LPV/R	-	-	-1	-1	0	0	-1
Gen444	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen448	AZT	3TC	IDV	RTV	-1	-1	-1	-1	-1
Gen460	3TC	EFV	LPV/R	-	1	-1	-1	0	-1
Gen480	AZT	3TC	IDV	RTV	1	1	-1	-1	-1
Gen503	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen54	D4T	DDI	IDV	-	-1	-1	-1	0	1
Gen571	DDI	3TC	IDV	RTV	-1	1	-1	-1	-1
Gen577	ABC	3TC	IDV	RTV	-1	-1	-1	-1	-1
Gen61	AZT	ABC	3TC	-	-1	-1	1	0	-1
Gen643	DDI	AZT	-	-	-1	-1	0	0	-1
Gen645	EFV	RTV	SQV	-	-1	-1	-1	0	-1
Gen667	DDI	3TC	IDV	RTV	-1	-1	-1	-1	-1
Gen668	D4T	ABC	IDV	RTV	-1	-1	-1	-1	-1
Gen670	LPV/R	SQV	-	-	-1	-1	0	0	-1
Gen671	AZT	3TC	-	-	-1	1	0	0	-1
Gen703	DDI	3TC	EFV	-	-1	-1	-1	0	-1
Gen75	EFV	IDV	RTV	-	-1	-1	-1	0	-1
Gen752	AZT	3TC	-	-	-1	-1	0	0	-1
Gen767	DDI	3TC	-	-	-1	-1	0	0	-1
Gen88	EFV	LPV/R	-	-	-1	-1	0	0	-1
Out2925861	LPV/R	SQV	-	-	-1	-1	0	0	-1
Out3601978	LPV/R	SQV	-	-	-1	-1	0	0	-1

The four single classifiers, the composite classifiers (DCC) including the TruGene system predict the drug susceptibility drug by drug. But in practice, doctor treats a combination of drugs to a patient. This combination is different depending on the symptoms of a disease of each patient and the doctor. Usually there are three or four drugs in a combination. Thus to evaluate the predictive performance of the four single classifiers and the proposed composite classifier (DCC) including the TruGene system with the clinical data, we design six rules for the drug combination prediction. These rules are defined as follow.

Rule 1 $\rightarrow$ If there is at least one drug that gives a prediction to resistant class, then a final prediction of that drug combination is resistant.

Rule 2 $\rightarrow$ If there are two drugs that give a prediction to resistant class, then a final prediction of that drug combination is resistant.

Rule 3 $\rightarrow$ If there are three drugs that give a prediction to resistant class, then a final prediction of that drug combination is resistant.

Rule 4 $\rightarrow$ If there is at least one drug that gives a prediction to susceptible class, then a final prediction of that drug combination is susceptible.

Rule 5 $\rightarrow$ If there are two drugs that give a prediction to susceptible class, then a final prediction of that drug combination is susceptible.

Rule 6 $\rightarrow$ If there are three drugs that give a prediction to susceptible class, then a final prediction of that drug combination is susceptible.

Table A.14 shows the accuracy of all models predicted by six rules. The results in Table A.14 demonstrate that rule 4 gave the best accuracy for all methods. Thus in further experiments, we used only the outputs from the rule 4 to compare the predictive performance among these methods.

Table A.14: The predictive accuracy of drug combination by six rules.

Rules	Accuracy (%)					
	TruGene	CBA	SVM	RBF Network	*k*-NN	DCC
rule 1	72.17	65.98	65.98	47.42	68.04	69.07
rule 2	79.38	81.44	80.41	78.35	80.41	81.44
rule 3	85.57	85.57	85.57	84.54	86.60	86.60
rule 4	**86.60**	**86.60**	**86.60**	**85.57**	**87.63**	**87.63**
rule 5	82.47	81.44	79.38	76.29	81.44	81.44
rule 6	77.32	75.26	72.17	60.83	75.26	75.26

Table A.15 shows the concordance between four single classifiers and the composite classifier with the TruGene system. The concordance calculated from the number of the samples with the same prediction from a pair of algorithms divided by the number of total cases for each drug.

Table A.15: The concordance between each model and the TruGene system.

Drug	Concordance with TruGene (%)				
	CBA	SVM	RBF Network	*k*-NN	DCC
LPV	97.22	**98.61**	**98.61**	**98.61**	**98.61**
APV	91.67	93.06	90.28	**94.44**	93.06
NFV	84.72	81.94	81.94	**88.89**	83.33
IDV	91.67	93.06	91.67	**94.44**	91.67
SQV	95.83	**97.22**	88.89	94.44	95.83
RTV	**95.83**	**95.83**	94.44	93.06	94.44
3TC	**87.50**	82.29	84.38	83.33	84.38
ABC	67.71	68.75	63.54	64.58	**69.79**
AZT	88.54	91.67	94.79	**95.83**	93.75
D4T	76.04	**80.21**	**80.21**	73.96	77.08
DDC	77.08	77.08	**79.17**	**79.17**	77.08
DDI	60.42	**69.79**	53.13	64.58	67.71
DLV	82.29	85.42	84.38	82.29	**86.46**
EFV	57.29	**91.67**	89.58	83.33	90.63
NVP	93.75	95.83	94.79	93.75	**96.88**
average	83.17	**86.83**	84.65	85.65	86.71

The results from Table A.15 show that the SVM model provided the highest average concordance (86.83%) to the TruGene system while the DDC model gave the second best average concordance. The third and the fourth best in average concordance methods were *k*-NN and the RBF network, respectively. Finally, the model which provided the lowest average concordance was CBA.

The comparisons of the predictive performance of four single classifiers, the proposed composite classifier (DCC), and the TruGene system with clinical data is illustrated in Talbe A.16. The number of the patients predicted to each class (clinical success or failure) of all methods is shown in the first two rows of the table. The row TP, TN, FP, and FN represent the number of true positive, true negative, false positive, and false positive clinical-data. The concordance is the concordance between other classifiers and the TruGene system. The concordance correct is calculated from the number of samples which are correctly classified by both algorithms divided by the number of the samples with the same prediction from those algorithms.

Table A.16: The comparisons of predictive performance among all methods.

Methods	TruGene	CBA	SVM	RBF Network	*k*-NN	DCC	Clinical
Clinical Success	96	96	96	93	97	97	85
Clinical Failure	1	1	1	4	0	0	12
Accuracy (%)	86.60	86.60	86.60	85.57	87.63	87.63	
TP (Clinic+,other+)	0	0	0	1	0	0	
TN (Clinic-,other-)	84	84	84	82	85	85	
FP (Clinic-,other+)	1	1	1	3	0	0	
FN (Clinic+,other-)	12	12	12	11	12	12	
Sensitivity (%)	0.00	0.00	0.00	8.33	0.00	0.00	
Specificity (%)	98.82	98.82	98.82	96.47	100.00	100.00	
Concordance (%)	-	97.94	97.94	94.85	98.97	98.97	
Concordance Correct (%)	-	87.37	87.37	88.04	87.50	87.50	

The results in Table A.16 show that all of models predicted the outputs more than 93 cases as clinical success, especially for *k*-NN and DCC models which totally gave predictions to clinical success. For the accuracy of all models, *k*-NN and DCC provided the best accuracy (87.63%). CBA and SVM gave the same accuracy with TruGene (86.60%), and the RBF network yielded the lowest accuracy (85.57%). Thought TruGene, CBA, and SVM predicted only one sample to clinical failure, these samples are wrongly classified (see the TP row). Thus the sensitivities of these models are zeroes. However the RBF network predicted four patients as clinical failure, and only one of these four patient was correctly classified (see the TP row), so the sensitivity of this model is only 8.33%. On the other hand, since *k*-NN and DCC predicted all patients as clinical success, these methods then provide 100% of specificities. For evaluating the concordance of the drug combination prediction between four single classifiers and our proposed composite classifiers with TruGene, we found that DCC gave the highest concordant relation (98.97%). While the RBF network provided the lowest concordant relation, this model yielded the best in concordant correct prediction (88.04%).

APPENDIX B

PUBLICATIONS

B.1 International Journal

1. Srisawat, A. and Kijsirikul, B. Combining Classifiers for HIV-1 Drug Resistance Prediction. Protein & Peptide Letters 15 (May 2008).

B.2 International Conference

1. Srisawat, A. and Kijsirikul, B. Using Associative Classification for Predicting HIV-1 Drug Resistance. Proceedings of the Fourth International Conference on Hybrid Intelligent Systems (HIS'04) (December 2004): 280-284.
2. Srisawat, A., Phienthrakul, T, and Kijsirikul, B. SV-kNNC: An Algorithm for Improving the Efficiency of k-Nearest Neighbor. Lecture Notes in Computer Science 4099 (August 2006): 975-979.
3. Srisawat, A. and Kijsirikul, B. MRBF: A Method for Predicting HIV-1 Drug Resistance. Proceedings of the Fourth International Conference on Intelligent Information Processing (ICIIP'06) (September 2006): 327-336.
4. Srisawat, A. and Kijsirikul, B. Combining Classifiers for HIV-1 Drug Resistance Prediction. The 2007 International Conference on Intelligent Computing (ICIC2007), China, August 21-24, 2007.

BIOGRAPHY

Name	Anantaporn Srisawat
Sex	Female
Marital Status	Single
Date of Birth	October 29, 1979
Place of Birth	Ratchaburi
Permanent Address	235 Moo. 11, Don Kruai, Damnoensaduak, Ratchaburi, 70130

Education:

2008 **Ph.D.** in Computer Engineering, Chulalongkorn University

Funding source:

- National Center for Genetic Engineering and Biotechnology
- Thai Government Scholarship

2003 **M.Sc.** in Computational Science, Chulalongkorn University

Funding source:

- Thai Government Scholarship

2000 **B.Sc.** in Computer Science, Silpakorn University