# ขั้นตอนวิธีทางคณิตศาสตร์เพื่อใช้ศึกษาโรคที่มีความซับซ้อน: กรณีศึกษาโรคบีตา$^0$-ธาลัสซีเมียฮีโมโกลบินอีและโรคโครน

นางสาวคันธารัตน์ อเนกบุณย์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรดุษฎีบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2552

# A MATHEMATICAL ALGORITHM TO STUDY THE COMPLEX DISEASES: A CASE STUDY OF BETA$^0$-THALASSEMIA/HB E'S AND CROHN'S DISEASES

Miss Khantharat Anekboon

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy Program in Computer Science
Department of Mathematics
Faculty of Science
Chulalongkorn University
Academic year 2009

| | |
|---|---|
| Thesis Title | A MATHEMATICAL ALGORITHM TO STUDY THE COMPLEX DISEASES: A CASE STUDY OF BETA$^{0}$-THALASSEMIA/HB E'S AND CROHN'S DISEASES |
| By | Miss Khantharat Anekboon |
| Field of Study | Computer Science |
| Thesis Advisor | Professor Chidchanok Lursinsap, Ph.D. |
| Thesis Co-Advisor | Suphakant Phimoltares, Ph.D. |

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

S. Hannongbua

.................................................... Dean of the Faculty of Science

(Professor Supot Hannongbua, Dr. rer. nat.)

THESIS COMMITTEE

Rajalida Lipikorn

.................................................... Chairman

(Assistant Professor Rajalida Lipikorn, Ph.D.)

C. Lur

.................................................... Thesis Advisor

(Professor Chidchanok Lursinsap, Ph.D.)

Suphakant Phimoltares

.................................................... Thesis Co-Advisor

(Suphakant Phimoltares, Ph.D.)

Suchart Chanama

.................................................... Examiner

(Assistant Professor Suchart Chanama, Ph.D.)

Suthat Fucharoen

.................................................... External Examiner

(Professor Suthat Fucharoen, M.D.)

.................................................... External Examiner

(Sissades Tongsima, Ph.D.)

คันธารัตน์ อเนกบุณย์ : ขั้นตอนวิธีทางคณิตศาสตร์เพื่อใช้ศึกษาโรคที่มีความซับซ้อน: กรณีศึกษาโรคบีตา$^0$-ธาลัสซีเมียฮีโมโกลบินอีและโรคโครน. (A MATHEMATICAL ALGORITHM TO STUDY THE COMPLEX DISEASES: A CASE STUDY OF BETA$^0$-THALASSEMIA/HB E'S AND CROHN'S DISEASES) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ศ.ดร.ชิดชนก เหลือสินทรัพย์, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม : ดร.ศุภกานต์ พิมลธเรศ 62 หน้า.

หลังจากโครงการจีโนมมนุษย์เสร็จสิ้นลง การศึกษาความสัมพันธ์ของกลุ่มผู้ป่วยกับกลุ่มควบคุมถูกนำมาใช้เพื่อสกัดความรู้ด้านความสัมพันธ์ของโรค ในขณะนี้ค่าใช้จ่ายและเวลาที่ใช้ในการหาจีโนไทป์ของสนิปส์ได้ลดลงกว่าอดีต การหาจีโนไทป์ของสนิปส์บางตัวหรือทั้งจีโนมเพื่อที่จะศึกษาความสัมพันธ์ของโรคสามารถทำได้ อย่างไรก็ตามการเพิ่มจำนวนสนิปส์ในการศึกษาความสัมพันธ์ของโรคจะส่งผลให้มีจำนวนคำตอบของตำแหน่งของยีนที่ไวกับการเกิดโรคที่เป็นไปได้ทั้งหมดเพิ่มขึ้นสูงอย่างมาก วิทยานิพนธ์ฉบับนี้ได้นำเสนอวิธีการเลือกตำแหน่งที่สำคัญสำหรับใช้ในการแบ่งกลุ่มใหม่ชื่อว่า IFGA พร้อมด้วย BoostMode-SVM ข้อมูลจีโนไทป์ของโรคบีตา$^0$-ธาลัสซีเมียฮีโมโกลบินอีและโรคโครนถูกนำมาทดสอบ ผลการทดลองพบว่าวิธี IFGA พร้อมด้วย BoostMode-SVM ให้ความถูกต้องมากกว่าทั้งในโรคบีตา$^0$-ธาลัสซีเมียฮีโมโกลบินอีและโรคโครน เมื่อเปรียบเทียบกับวิธี Optimum Random Forest และ CART โรคบีตา$^0$-ธาลัสซีเมียฮีโมโกลบินอีใช้ 6 ตำแหน่งในการแบ่งกลุ่มระหว่างกลุ่มผู้ป่วยกับกลุ่มควบคุมด้วยความถูกต้อง 71.57% และใช้ 8 ตำแหน่งสำหรับการแบ่งกลุ่มระหว่างกลุ่มผู้ป่วยกับกลุ่มควบคุมของโรคโครนด้วยความถูกต้อง 71.06% ด้วยการเฉลี่ยจากการทดสอบ 10 ครั้ง

|  |  |
|---|---|
| ภาควิชา คณิตศาสตร์ | ลายมือชื่อนิสิต |
| สาขาวิชา วิทยาการคอมพิวเตอร์ | ลายมือชื่ออ.ที่ปรึกษาวิทยานิพนธ์หลัก |
| ปีการศึกษา 2552 | ลายมือชื่ออ.ที่ปรึกษาวิทยานิพนธ์ร่วม |

# # 4773855023 : MAJOR COMPUTER SCIENCE

KEYWORDS : FEATURE SELECTION / SUPPORT VECTOR MACHINE / GENETIC ALGORITHMS / CASE-CONTROL ASSOCIATION STUDY / BIOINFORMATICS

KHANTHARAT ANEKBOON : A MATHEMATICAL ALGORITHM TO STUDY THE COMPLEX DISEASES: A CASE STUDY OF BETA$^0$-THALASSEMIA/HB E'S AND CROHN'S DISEASES. THESIS ADVISOR : PROF.CHIDCHANOK LURSINSAP, Ph.D., THESIS CO-ADVISOR : SUPHAKANT PHIMOLTARES, Ph.D., 62 pp.

After the human genome project, case-control association studies have been used to extract knowledge of disease association from SNPs. Costs and time in genotyping SNPs are reduced, genotyping SNPs or whole genome is now possible to study the association in a particular disease. However, increasing a number of SNPs affects a number of all possible cases which grows exponentially. A new feature selection and classification called IFGA with BoostMode-SVM is proposed. Two real data sets of case-control association study of Beta-0/Hb E Thalassemia and Crohn's disease from a given set of genotype data are evaluated. The IFGA for feature selection with the BoostMode-SVM classification performs well in both Thalassemia and Crohn's diseases compared with the previous techniques: Optimum Random Forest and CART. We used 6 features as biomarkers for Thalassemia with 71.57% accuracy and 8 features for Crohn's disease with 71.06% accuracy by 10-fold cross validation.

Department : Mathematics Student's Signature Khantharat Anekboon

Field of Study : Computer Science Advisor's Signature

Academic Year : 2009 Co-Advisor's Signature Suphakant Phimoltares,

# Acknowledgements

I would like to express my deepest gratitude to my advisor Prof. Dr. Chidchanok Lursinsap for great advises, a lot of valuable ideas, and encouragements throughout my dissertation and my Ph.D. program. I would also like to extend my appreciation to Dr. Suphakant Phimoltares, my co-advisor, for his valuable suggestion, spending time discussion and review my papers.

I would like to thank the Office of Higher Education Commission for their financial support. I am also very grateful to Prof. Suthat Fucharoen for his kindness of issuing the research objective of this dissertation and giving the data for the experiment. I would like to extend my sincere appreciation to Dr. Sisades Tongsima for sharing his valuable suggestion and giving helpful advices.

Special thanks to Miss Wattanan Makarasara and Miss Sasiporn Tongman for the biological knowledge. Thanks a lot to all AVIC members especially Mr. Peerapol Khunarsa, Mr. Preecha Tangkraingkij, Mr. Narongsak Putpuek, Mr. Jirawat Chaiareerat, Miss Putthiporn Thanathamathee, Mr. Wittawin Susutti, Mr. Juthawut Chantharamalee, and Mr. Jitti Boonkueng for kindness giving me their computers to run programs.

Last but not least, I would like to express my sincere gratitude and deep appreciation to my parents, my family, and my friends for constant encouragement, love, and supports throughout my life.

# CONTENTS

# List of Tables

# List of Figures

# CHAPTER 1

# INTRODUCTION

## 1.1 Introduction and Problem Review

As a result of the human genome project, many researches aim to extract knowledge from the nucleotide sequences. Which genes affect a disease is an interesting question. To answer this problem, association study has been used to identify disease susceptibility loci associated with a particular disease [1]. Normally, disease can be categorized into common disease and complex disease. Common disease is a single gene disorder as the result of gene inheritance. On the other hand, the complex disease involves more than one genes with complex relationship affecting to a disease [2, 3]. Although we already knew which gene affects the Thalassemia's disease, those genotypes cannot classify the phenotype correctly. Consequently, only one gene may not be sufficient to categorize the phenotype. For Crohn's disease, even though the susceptibility loci for the causative were report, those known loci do not completely explain the genetic risk. Therefore, genome-wide association study is a contribution to identify the additional loci [4].

This research aims to discover the significant features at the molecular level which associate to beta$^0$/Hb E Thalassemia's and Crohn's diseases. Association study is usually investigated by SNPs [5]. Single Nucleotide Polymorphism (SNP), the most frequent form of genetic variations [6], is a single base substitution from one nucleotide to another one. For instance, the first DNA sequence is 'AACTGCGTT' and the other is 'AGCTACGAT'. The SNPs are in the second, fifth, and eight loci of the sequences. In general, SNPs are variation bases from comparing nucleotide sequences at the same locus. For current estimation, SNPs occur as frequently as every 100-300 bases of a sequence [7]. 99.9 percent of human genome sequences are identical, while only 0.1 percent difference. In other words, each person has the unique SNPs. Therefore, SNPs can function as genomic markers for identifying the people. With this assumption, some SNPs should be the efficient identifier for association study as well.

Due to recent technology, cost and time in finding SNPs are reduced. We can now genotype a large numbers of SNPs. However, genotyping more SNPs results more dimensions of data which causes time consuming in extraction of the significant biomarkers. It is not practical to genotype a large number of SNPs to create a model of association study. Therefore, feature selection should be considered in this manner. Feature selection [8] is a technique of selecting subset of relevant features or attributes. This technique provides more effective in the point of increasing learning accuracy and helps us to understand the significant of features by removing irreverent or redundant features. Feature selection has been widely used in almost fields, including bioinformatics. In general, feature selection can be categorized into three categories - filter, wrapper, and embedded: depending on whether the classifier is in the model. Firstly, filter method does only feature selection part without communicating with the results from the classifier. The advantage of this method is computing speed but it does not guarantee high classification accuracy. Secondly, wrapper method includes the classification module into a model. Due to the feature evaluation, this method is more effective than filter method. However, wrapper method takes more time than the previous one because the classifier is performed every time features are selected. Thirdly, for the embedded method, feature selection search and classification are run simultaneously.

In another point of view, feature selection technique can be divided into two groups depending on a technique of selecting variables, univariate and multivariate paradigm. Univariate paradigm simply selects features depending on their own property. Feature scores are ranked according to a criterion from most to least significance. After that, subset of ranked features was evaluated to discover the most informative subset [9]. This method is simple and fast. However, it does not concern the interaction or dependence of variables. Examples for this technique are chi-square and t-test. On the contrary, because multivariate paradigm considers groups of features together so it concerns the interaction between features. Examples for this technique are MDR, PAM, and genetic algorithm. Multivariate is an attractive method because, in the real world disease, we have a model of incomplete penetrance or interaction. For example, a feature itself does not have a marginal effect whereas combination of features gives high effect to a particular disease. Since multivariate paradigm covers more possible cases than univariate (interaction, in this case), multivariate paradigm should give more effective results than a univariate paradigm. It has been noted that in molecular genetics, interaction can be called as epistasis. Epistasis concentrates on interactions among a group of features since some subsets of features may be important than individual one (marginal effect). Thereby, multiple variable analysis should be preferred.

Although multivariate paradigm provides an effective result, it is an exhaustive task if all possible subsets of features are performed. For *L* features, there are $2^L$ possible cases which exponentially grows with respect to *L*. Note that, it depends on the size of *L*. If the size is not large, generating all cases to search the best results is possible. However, searching all of cases is impossible for large *L*.

Another point of view of association study is the data type, genotype or haplotype. Although haplotypes have more meaning in biological task than genotypes, inferring haplotypes from genotypes gives ambiguous output in unphase genotype data especially in a case-control study. For example, if a person has genotype AT TT CG GA TC CC, this person may have haplotype one from these groups: ATCGTC/TTGACC, ATCGCC/TTGATC, ATCATC/TTGGCC, ATCACC/TTGGTC, ATGGTC/TTCACC, ATGGCC/TTCATC, ATGATC/TTCGCC, and ATGACC/TTCGTC. However, it is not guarantee which haplotype this person has. Moreover, it takes time to generate all possible haplotypes.

### 1.2 Statement of the Problem

Besides monogenic disorders, other genes can be involved to make considerable clinical variability between patients [10]. To discover the relation between genetic and the symptom, these questions arise:

1. What are the informative SNPs that can classify the patients?
2. How to reduce the irrelevant SNPs?

### 1.3 Research Objectives

The objectives of this dissertation are:

1. Identify the informative SNPs that can classify the clinical beta$^0$/ Hb-E Thalassemia patients.
2. Identify the informative SNPs that can classify the clinical Crohn's patients.
3. Minimize those SNPs while the accuracy of classifying is still good.

### 1.4 Scopes of the Study

The scopes of work in this dissertation are:

1. The proposed algorithms are for searching the informative SNPs for classification the clinical patients of beta$^0$/Hb E Thalassemia.
2. The proposed algorithms are for searching the informative SNPs for classification the clinical patients of Crohn's disease.
3. The input for proposed method is missing value free.

# CHAPTER 2

# THEORIES AND LITERATURE REVIEWS

This chapter briefly illustrates related concepts applied in this dissertation: the basic concepts of biological background, genetic algorithms, bootstrap, and support vector machine. Literature relating to association study of the case/control data is also reviewed.

## 2.1 Biological Background

The overview of human genetic is demonstrated in Figure 2.1. Cell is the smallest living unit of all living organisms. A nucleus of each cell contains the set of chromosomes, where each chromosome is composed of sequences of the DNA (deoxyribonucleic acid). DNA has four chemical bases: Adenine (A), Cytosine (C), Guanine (G), and Thymine (T). Two bases are paired via hydrogen bonds, where adenine is paired with thymine and cytosine is paired with guanine.

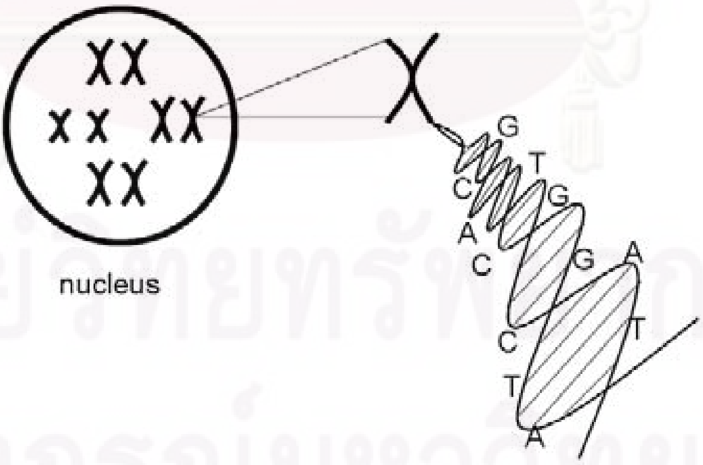

Figure 2.1 The overview of genetic.

Association study is a study aiming to discover an association between genetic polymorphisms and phenotype [11]. Genetic polymorphisms of the association study may be in a form of genotype or haplotype. In diploid organisms such as human, each chromosome has two homologous copies. Sequence of nucleotides in the same chromosome is called haplotype whereas the nucleotide at the same locus of those two copies of each chromosome is called genotype. For example, the haplotypes of the first DNA sequence are AACCTAGCAC and TTGGATCGTG. The haplotypes of the second DNA sequence are AACTTAGCAC and TTGAATCGTG. On the other hand, the genotypes of these sequences are AA, AA, CC, CT, TT, AA, GG, CC, AA, and CC, ordered from left to right respectively. It has been noted that, there is the difference between two chromosomes in the fourth locus. This locus is the position of the SNP.

## 2.2 Genetic Algorithms

A genetic algorithm is a search technique which applies from the principle of genetic. The genetic algorithm maps the real world problem into a into a chromosome-like data structure. In general, a framework of genetic algorithm can be demonstrated in Figure 2.2. A searching problem is an input of the genetic algorithm. To begin with, encoding is required for translating the real world problem into a chromosome-like structure. Then, generate an initial population of a generation. It has been noted that, each chromosome corresponds to candidate solution and the population consists of a set of chromosomes. After that, all chromosomes in this generation are evaluated; the good chromosome, which represents a good solution to a target, is obtained to use in the next processes. Then the chromosome is iteratively applied by recombination, crossing over, mutation to create the next population. Therefore, good chromosome has more chance to re-produce in the next population.

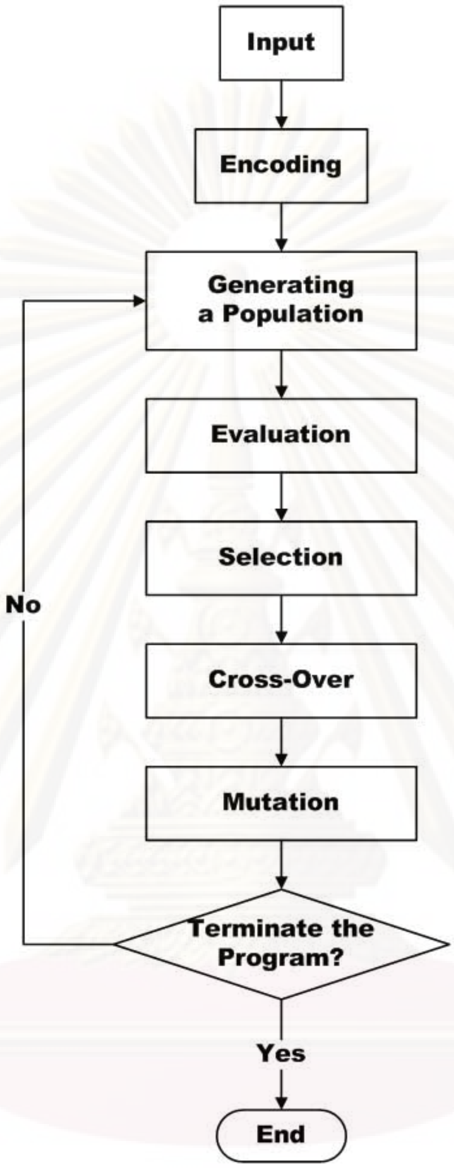

Figure 2.2: The genetic algorithm framework.

### 2.2.1 Chromosome Encoding

The chromosome in the genetic algorithm can be encoded in different ways. It can be divided into two groups, a binary and a real value encodings. The binary encoding encodes data into either 0 or 1 in each position while the real value encoding encodes data into a string of real numbers. Values of an encoded data can be arbitrary depending on the type of problem such as {A, C, C, G, A} or {14.2, 31, 8, 22.7, 5} **[12]**.

### 2.2.2 Generating the Population

The population in each generation consists of a set of chromosomes, where a number of chromosomes in the generation is called the population size. In every chromosome, the length of chromosomes must be the same in all generation. Each value in the chromosome is called an allele. For example, Figure 2.3 has 5 population in this generation and the length of chromosome is 5. Generating the population can divide into 2 steps: 1) initially define the population and 2) generate the population for the next generation. Initializing the population of the first generation may randomly generates a set of chromosomes but the population of the next generation must be fed to the processes of cross-over and mutation.

| Chromosome No. | Population |
|---|---|
| 1 | 1 0 0 1 1 |
| 2 | 0 0 0 0 1 |
| 3 | 1 1 0 1 0 |
| 4 | 1 0 1 1 1 |
| 5 | 0 0 0 0 1 |

Figure 2.3: Population in current generation.

### 2.2.3 Evaluation

The chromosome is the candidate solution of searching problem. An evaluation is the process of measure fitness of each individual chromosome. A fitness function returns the fitness value determining how close the chromosome and the solution. For example, suppose that the searching problem is finding features used for a classification problem. The fitness function may be the number of miss-classification.

### 2.2.4 Selection

Selection is a process of selecting good parents into a mating pool to re-produce offspring of the next generation. Selection is a first step in generating a next generation. It uses a principle of survival, better fitness score means more chance to survive in a next generation. Each individual chromosome was selected based on its fitness score into a mating pool, which there are several algorithms such as roulette wheel, tournament, rank, steady-state, and elitism.

### 2.2.5 Cross-Over

An idea of crossing over in a GA is like cross-over in biology genetic. Parents will be random selected from a mating pool. Then, chromosomes from parent produce new offspring's chromosomes for a next generation via a cross-over process. A basic idea of cross-over is swapping chromosome values between two parents from a cross-over point to the end of a chromosome or to the next cross-over point. For example, in binary encoding, $\text{parent}_1$ has a chromosome 101 * 00 while $\text{parent}_2$ has 110 * 11 where * represents a cross-over point. Then, offspring chromosomes are 10111 and 11000.

### 2.2.6 Mutation

A mutation is changing value of a specified locus. This specified locus is called a mutation point. Again, an idea of mutation in a GA is like mutation in biology genetic. DNA bases can be changed into other bases when mutation occurs. Like mutation in the genetic algorithms, it changes the value of the specified locus to other values. For instance, a chromosome is encoded into a binary as 10111 and the mutation point occurs at the first position, then this chromosome is changed to 00111.

### 2.2.7 Termination

After a next generation is generated, the process of calculation of a fitness score is then executed. These processes are recursively done until a terminal condition is met. The termination condition can be a number of generation, a time limitation, a number of the same consecutive results, etc.

## 2.3 Bootstrap

A concept of bootstrap is estimating parameters by re-sampling from the original data. This is an attractive technique when only few data are available and inapplicable to sample more data. Figure 2.4 is a diagram of bootstrap method. Let $\hat{\theta} = s(x)$ be an interested parameter with an unspecified distribution *F*. Let $x = \{x_1, x_2, \ldots, x_n\}$ is a collection of *n* individual real-world random sampling data. Each $x_i \in x$ where $i = 1, 2, \ldots, n$ is independent and identically distributed sample from *F*. In the bootstrap world, $\hat{\theta}^{*j} = s(x^{*j})$ be an interested parameter of bootstrap $j^{th}$ with an empirical distribution $\hat{F}$. Noted that, $j^{th}$ is a running number from 1 to the number of bootstrap sample, *B*. A bootstrap sample $x^{*j} = \{x^{*j}_1, x^{*j}_2, \ldots, x^{*j}_n\}$ is generated by randomly sampling *n* times with replacement from the original real-world sampling *x*.

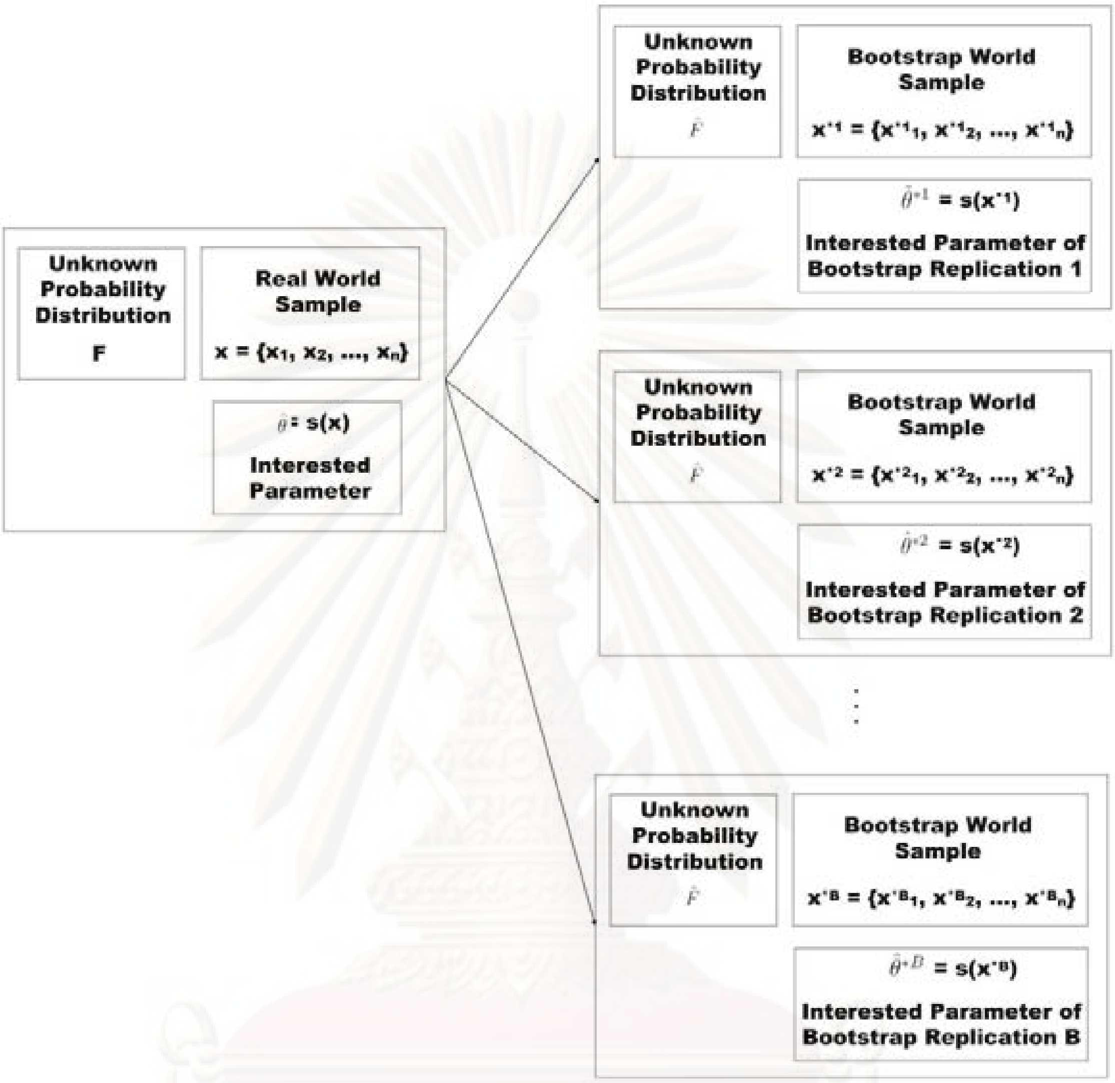
Unknown Probability Distribution
F
Real World Sample
x = {x1, x2, ..., xn}
θ̂ = s(x)
Interested Parameter
Unknown Probability Distribution
F̂
Bootstrap World Sample
x*1 = {x*11, x*12, ..., x*1n}
θ̂*1 = s(x*1)
Interested Parameter of Bootstrap Replication 1
Unknown Probability Distribution
F̂
Bootstrap World Sample
x*2 = {x*21, x*22, ..., x*2n}
θ̂*2 = s(x*2)
Interested Parameter of Bootstrap Replication 2
Unknown Probability Distribution
F̂
Bootstrap World Sample
x*B = {x*B1, x*B2, ..., x*Bn}
θ̂*B = s(x*B)
Interested Parameter of Bootstrap Replication B

Figure 2.4 The Bootstrap diagram.

### 2.4 Support Vector Machine

The SVM [13] is a widely used learning machine for a two-group classification problem. SVM classifier classifies its data set into only two classes by using a hyperplane without adjusting weights as other models such as RBF network and multilayer perceptron network with backpropagation learning. The hyperplane is used as a linear decision surface to classify two groups of data. The linear decision surface is the maximum margin of separation between two classes. Input vectors can be both linear and non-linear separable. For nonlinear version, a kernel function is required to map input vectors into a high-dimension feature space before constructing a linear decision surface. The separable of SVM is based on the capability of the kernel mapping function which maps the given data set into another higher dimensional space. A good kernel function can relocate all data set from different classes in the new dimension such that there is no overlap of data between classes.

Each training patterns consist of $\{\boldsymbol{x}_i, y_i\}$ where $\boldsymbol{x}_i$ is a $i^{th}$ input data and $y_i$ is a target of input $i^{th}$ such that $y_i \in \{-1, 1\}$. The data are linearly separable if there exists a weight vector ***w*** and a scalar bias *b* such that

$$y_i\,(w^T \cdot x_i + b) \geq 1,\ i = 1, \dots , n \qquad (2.1)$$

The decision function is defined as follows.

$$y_i = \text{sign}\,(w^T \cdot x_i + b) \qquad (2.2)$$

However, if the data are non-linearly separable, the transformation from ***n***-dimensional input vector ***x*** into ***m***-dimensional feature vectors, for $n < m$, is needed. Suppose $\varphi(x)$ denotes a non-linear transformation from an input space into a feature space. Thus, there exists a weight vector ***w*** and a scalar bias *b* such that

$$y_i\,(w^T \cdot \varphi(x_i) + b) \geq 1,\ i = 1, \dots , n \qquad (2.3)$$

The decision function for non-linearly separable data becomes

$$y_i = \text{sign}\,(w^T \cdot \varphi(x_i) + b) \qquad (2.4)$$

The goal of the SVM is finding a hyperplane with the maximum margin, equation (2.5), of separation of two classes data points where this optimization problem subjects to equation (2.1) for the linear separable case and subjects to equation (2.3) for the non-linear separable case.

$$\min \frac{1}{2}\|w\|^2 \qquad (2.5)$$

The weight vector ***w*** is calculated by the Lagragian function equation (2.6) for linear separable case or equation (2.7) for non-linear separable case.

$$L(w,b,\alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i \left[ d_i (w^T \cdot x_i + b) - 1 \right] \qquad (2.6)$$

$$L(w,b,\alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i \left[ d_i (w^T \cdot \varphi(x_i) + b) - 1 \right] \qquad (2.7)$$

The Lagrance function can be translated into a dual problem by maximize the objective function, *L* in equation (2.8) where $\alpha$ is called Lagrange multiplier. Then ***w*** can be calculated as equation (2.9).

$$L(w,b,\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j x_i^T x_j \qquad (2.8)$$

$$\text{Subject to } \sum_{i=1}^{N} \alpha_i d_i = 0 \text{ and } 0 \leq \alpha_I \text{ for } i = 1, 2, \ldots, N$$

$$w = \sum_{i=1}^{N} \alpha_i d_i x_i \qquad (2.9)$$

The SVM technique above is called hard margin optimization. The main problem of the hard margin is that it always produces no training error, as equation (2.1), no data points falls in the margin of separation. When a model does not allow some training error, it is sensitive to noise. Another technique of SVM is soft margin optimization.

The different margin of separation between hard and soft is a number of data falls in the margin of separation. Soft margin optimization allows some data points falls in the region with a particular number of errors for non-separable data. Equation (2.1) is rewritten for soft margin to equation (2.10), where $\xi_i$ is called a slack variable. The slack variable represents the deviation of the data point from the correct classification.

$$y_i (\mathbf{w}^T \cdot \mathbf{x}_i + b) \geq 1 - \xi_I, \text{ where } j = 1, \ldots, n \text{ and } \xi_i \geq 0 \qquad (2.10)$$

Since some misclassification error are allowed in the soft margin, the goal of the SVM is changed to finding a maximum separated hyperplane with the minimum misclassification error equation (2.11), where *C* is the user defined parameter which tread off between the number of error and the margin.

$$\min\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \tag{2.11}$$

$$L(w,b,\xi,\alpha,\beta) = \frac{1}{2}w^T w - C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\beta_i\xi_i \tag{2.12}$$

$$-\sum_{i=1}^{N}\alpha_i\left[d_i(w^T\cdot x_i + b) - 1 + \xi_i\right]$$

$$L(w,b,\xi,\alpha,\beta) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j x_i^T x_j \tag{2.13}$$

Subject to 1 $\alpha_i\left[d_i(w^T\cdot x_i + b) - 1 + \xi_i\right] = 0$ for i = 1, 2, ... , N

2 $\beta_i\xi_i = 0$ for i = 1, 2, ... , N

3 $\alpha_i \geq 0$ and $\beta_i \geq 0$ for i = 1, 2, ..., N

$$\mathbf{w} = \sum_{i=1}^{N_S}\alpha_i d_i \mathbf{x}_i \tag{2.14}$$

The Lagrance function of soft margin optimization is equation (2.12) where $\alpha$ and $\beta$ are Lagrange multipliers. The dual objective function is equation (2.13). Then ***w*** can be calculated as equation (2.14) where *Ns* is the number of the support vectors. It has been noted that, for the non-linear separable case, the data point $x_i$ from equation (2.12), (2.13), and (2.14) is changed to $\varphi(x_i)$.

## 2.5 Literature Review

The objective of an association study research is to identify loci for a particular disease. Previous technique for case-control association studies are discussed below.

### 2.5.1 Chi-Square

Chi-square test for independence was used to test for association with the disease trait to identify of susceptibility loci for complex diseases in a case-control association study using the genetic analysis workshop 14 dataset [14]. The paper used this technique with disease that does not follow simple Mendelian inheritance. The aim is to determine the ability of conventional association methods to fine map a locus of interest. 14 Simulated datasets were taken from the Genetic Analysis Workshop then were analyzed for association between the disease traits and locus D2. An association between disease and a locus was test by Pearson’s chi-square tests for independence. However, this method gives the low significance levels because no association was detected for SNPs which to be linked to the disease locus.

### 2.5.2 Classification and Regression Tree

A multigenic approach using a classification and regression tree (CART) predicts breast cancer risk was proposed [10]. The aim of this paper is to find the multigenic significant genes for classifying the breast cancer risk from case and control. CART is a non-parametric technique using decision tree which can be used for data mining. The classification tree recursively partitions data into 2 subsets, case and control. The algorithm starts with searching 1 variable at a time to divide the whole data into 2 subgroups. The selected variable is the variable that maximize the purify data into 2 groups. Subsequently, it repeats finding another variable for splitting those data into 2 more subgroups. This process iterates until the stopping criteria reaches or data cannot be divided into 2 subgroups which is called terminal node. In this paper, trees are constructed by training data with 10 fold cross validation. For each tree, CART constructs the maximal tree possible and then prunes some branch of tree which contributes least to overall accuracy. Only the best tree was selected for further analysis. However, this method has to consider only 1 feature at a time. What if more than 1 position is equivalent importance? Moreover, tree can be constructed with an ambiguous, which means than there are more than 1 tree that give the same accuracy of prediction.

### 2.5.3 Multifactor-Dimensionality Reduction

A multifactor-dimensionality reduction (MDR) **[15]** is the dimension reduction method for multilocus. The objectives of MDR are (1) to reduce the dimensionality of multilocus information and (2) to identify the combination of polymorphism which associates with disease risk. The MDR technique is nonparametric and model free; so, it neither estimates any parameters nor assumes any particular genetic model. With MDR, multilocus genotypes are pooled into high-risk and low-risk groups so dimensions were reduced from n dimensions to 1 dimension. In n factors, MDR counts the number of all possible combining genotypes for case and control then fills into n dimensional space. For example, for 2 loci (A/a and B/b), there are 3 possible genotypes each i.e., $x_i = \{AA, Aa, aa\}$, $y_j = \{BB, Bb, bb\}$. Therefore, there are 9 combinations of 2 locus genotype i.e., $x_i\, y_j$ such that $1 \leq i, j \leq 3$.

In the space, each cell is classified into high-risk or low-risk by comparing the ratio of case and control with some threshold. 10-fold cross-validation was used to find the average prediction error of each model. Then the model with minimum error was chosen. However, this approach uses all combination of genotypes, which they may exceeds computational feasibility when they are huge number of features.

### 2.5.4 Optimum Random Forest

An Optimum Random Forest (ORF) **[16]** was proposed to predict the genetic susceptibility of complex diseases. Let *M* be a number of all features and *m* be a number of selected features defined by user. The ORF algorithm starts with randomly generating a set of *MSC*, the set of the position of selection feature. The length of each $MSC_i$ is *m* as well. Each $MSC_i$ must contains the position of feature $i^{th}$. Therefore, the size of *MSC* equals *M*. After the *MSC* set is randomly generated, each $MSC_i$ is used in generating *p* classification trees by permuting the *MSC* 's order and the training set. The value of *p* is defined by user. Test set is run down to the classification tree to get the accuracy rate. Therefore, each $MSC_i$ has *p* accuracy values and *p* classification trees. The algorithm chooses the best tree for each $MSC_i$ by selecting the tree with the highest accuracy of the test set. To find the important feature, all accuracy value of the same feature of the *MSC* is sum up. The highest score means the most important feature. The final *m* selected features (*MDMSC*) are selected from this important score.

To predict the test set, the *MDMSC* value is re-ordering and then, re-generating a tree $T_i$ for a particular number of re-ordering. For each $T_i$, the ORF algorithm generates *b* bootstrapped samples and generates *b* classification trees from $T_i$, where *b* is defined by user. The final tree for predicting the output of the test set is the tree with the highest average accuracy from *b* bootstrap samples.

# CHAPTER 3

# PROPOSED METHOD

In this section, we introduce a new encoding method called Integer encoding of the Feature selection in the Genetic Algorithm (IFGA). A summary of the IFGA method is demonstrated in Figure 3.1. The input data is a case-control data obtained in this study are in forms of sequences of genotypes of SNPs.

The first population, a set of chromosomes, is initiated by the integer encoding approach. The data in the chromosome represents a set of selected features. After the population is generated, each chromosome in this population is evaluated by the fitness score. This score is calculated from the BoostMode-SVM approach. Then, the IFGA re-generates the next population by IFGA selection, IFGA cross-over, and IFGA mutation until a termination criterion is met. The result from this approach is the selected features which have the highest accuracy in classification. The details about the IFGA encoding, IFGA generating the population, BoostMode-SVM, IFGA selection, IFGA cross-over, IFGA mutation, and termination are described in the following sections.

## 3.1 Integer Encoding Method

Dealing the genetic algorithm with feature selection method is usually encoded by binary encoding [17]. In binary encoding for feature selection, 1 represents selected feature and 0 represents unselected feature. The length of a chromosome equals a number of all features. Thereby, for a large number of features, a chromosome size is also large. With a large number of features, encoding by binary method uses a large number of bits. This is an important problem because of two reasons. First, the running time highly depends on the length of chromosome. Therefore, a long chromosome takes a long time to compute. Second, a general binary encoding does not fix a number of selected features. It fixes only the length of the chromosome. For example, encoding with binary method of the chromosome length 10, the first chromosome may be

0110101010, where 1 represents the selected feature. So, the selected features from the left side of the first chromosome are features 2nd, 3rd, 5th, 7th, and 9th. Therefore, the first chromosome selects 5 features whereas the second chromosome may be 0001010000 whose only 2 features are selected (feature 4th and feature 6th). It can be seen that a number of selected features varies from 1 to a number of all features, 10 in this case. Hence, it takes a lot of times to find an optimum features.

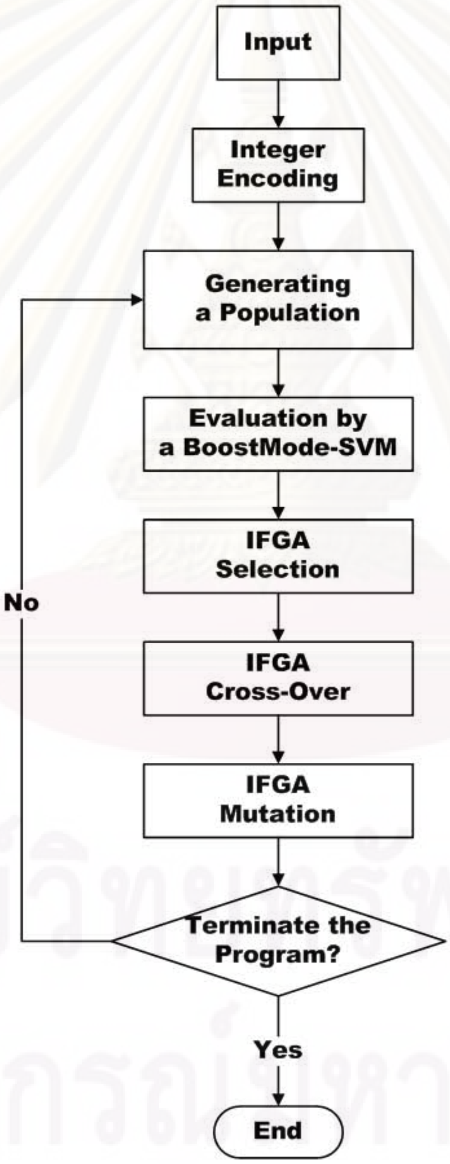

Figure 3.1: The overall IFGA flow chart.

The IFGA is proposed for solving those problems. A case-control data obtained in this study are in forms of $N$ genotype sequences. Each chromosome's length must be less than or equal to $M$, where $M$ is a number of user defined features. Our IFGA does not set every chromosome size equal the number of feature, $|N|$ like the binary encoding approach. Only few chromosome sizes are selected from the position of $M$ in the algorithm. First, the length of a chromosome to be processed in the genetic algorithm must be defined. Let $Q_i$ be the i$^{th}$ chromosome processed in the algorithm. The length of $Q_i$, denoted by $|Q_i|$, is set to a constant less than or equal to $M$, i.e. $|Q_i|$, < $M$. Then, a set of $|Q_i|$ random numbers whose values are less than or equal to $M$ are generated. These random numbers are used as the locations to select the corresponding genotypes from a given feature sequence. During the genetic algorithm, the length of each chromosome is not necessarily equal. For example, suppose the following sequences as shown in Figure 3.2 are given. The following number of locations is selected from each given sequence: 3 for the first sequence, 5 for the second sequence, and 1 for the third sequence.

| Seq. No. | Locations | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | CT | CC | GG | TA | TG | AT | AA |
| 2 | CC | CA | GT | TT | TT | AA | AC |
| 3 | TT | AA | GT | AA | TG | TT | CC |

Figure 3.2: An example of three given genotype sequences and locations.

| Seq. No. | Selected Locations | | | | | | | Chromosome in GA |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| 1 | CT | - | - | - | TG | AT | - | $Q_1 = \{5, 6, 1\}$ |
| 2 | CC | - | - | - | TT | AA | - | |
| 3 | TT | - | - | - | TG | TT | - | |
| 1 | CT | CC | - | TA | TG | AT | - | $Q_2 = \{4, 2, 1, 6, 5\}$ |
| 2 | CC | CA | - | TT | TT | AA | - | |
| 3 | TT | AA | - | AA | TG | TT | - | |
| 1 | - | - | - | - | - | - | AA | $Q_3 = \{7\}$ |
| 2 | - | - | - | - | - | - | AC | |
| 3 | - | - | - | - | - | - | CC | |

Figure 3.3: An example of three given genotype sequences: selected locations and the corresponding genotypes in each sequence.

There are three given sequences of length 7 (Figure 3.2). Here, $M$ is equal to 7. For the first given sequence (Figure 3.3), $|Q_1|$ is set to 3 and the randomly selected locations are 1, 5, and 6 (the chromosome $Q_i = \{1, 5, 6\}$). For the second given sequence, $|Q_2|$ is set to 5 and $Q_2 = \{4, 2, 1, 6, 5\}$. For the third given sequence, $|Q_3|$ is set to 1 and $Q_3 = \{7\}$. Figure 3.3 summarizes the corresponding genotype at each selected location.

### 3.2 Generating a Population

There are two kinds of population, initial population and the population of the next generation, called next population.

To generate the initialized population with *P* chromosomes, where *P* is user-defined number of chromosome in the population, there is no element in the population set. Then, repeatedly generates the chromosome and add it into the set of population until the number of the chromosome in the population set equals to *P*. The processes of generating the chromosome are the processes of the integer encoding of the chromosome, which previously described. After the chromosome *Q* is generated, this chromosome is added into the population set.

On the other hand, the population in a next generation consists of

- the best fitness score, *b*, from a current generation that survives in the next generation.
- *e* groups of features from evolution, cross-over and mutation.
- *r* groups of features from new re-selected features.

After *b* and *e* are added to the next generation, those chromosomes are checked for redundancy. Each chromosome must be identical in the next generation. Other duplicated chromosomes will be removed. If a number of chromosomes in the next generation is less than a number of chromosomes in current generation then new subsets of features, *r*, will be randomly created and added to the next generation.

### 3.3 IFGA Selection

There are three steps of apopulation evaluation: selection, cross-over, and mutation. Selection uses a principle of survival. A good fitness score means more chances to survive in a next generation. Each individual chromosome is selected based on its fitness score into a mating pool by a stochastic universal sampling method

(SUS) [18]. It also guarantees that the best chromosome in a current generation will be a chromosome in a next generation by an elitism technique.

The elitism method is a process that takes the best chromosome or some of the best chromosomes to the next generation. These best chromosomes will not be done in a cross-over and mutation processes [19]. Here, the technique of SUS is adapted.

Let $P$ be a set of considered chromosomes and $T$ a set of chromosomes selected from $P$. Generally, the size of $T$ is less than the size of $P$. Let $U$ be a set of pointers such that $|U| = |T|$. Consider the example in Figure 3.4. Suppose there are six survived chromosomes ($|T| = 6$) from eight chromosomes $|P| = 8$. The numbers 1 to 8 in Figure 3.4 are the chromosome number. SUS randomly selects a starting point, $U_1$, in a range of $\left[0, \frac{1}{|\mathrm{T}|}\right]$ [20]. Note that $U_1$ is the first element added into the set $U$. From this example, the starting point ($U_1$) is 0.16. The next ($\mathrm{U_i}$), $2 \leq \mathrm{i} \leq |\mathrm{U}|$, elements will be added into set $U$. Their values are calculated from $\mathrm{U_{i-1}} + \frac{1}{|\mathrm{T}|}$. Therefore, the set of $U$ becomes {0.16, 0.327, 0.494, 0.661, 0.828, 0.995}. The FNS is a set of scores for each chromosome $P_i$, where $i = 1,...,|P|$. This score is calculated from equation (3.1). Each $FNS_i$ value is used for setting the size of each chromosome block in the figure. Therefore, $FNS$ = {0.24, 0.45, 0.61, 0.74, 0.84, 0.92, 0.97, 1}. Note that the $fitness\_score_i$ is calculated by the BoostMode-SVM approach.

$$\mathrm{FNS_i} = \frac{\sum_{j=1}^{i} \mathrm{F_j}}{\sum_{i=1}^{|P|} \mathrm{F_i}};\ \mathrm{i} = 1, \ldots, |\mathrm{P}| \qquad (3.1)$$

Where $F_i = fitness\_score_i$

The pointers $U_i$, $1 \leq i \leq |\mathrm{T}|$ points at the survived chromosomes. Since $U$ = {0.16, 0.327, 0.494, 0.661, 0.828, 0.995} point to the chromosome numbers 1, 2, 3, 4, 5, and 7, respectively, the chromosomes 1, 2, 3, 4, 5, and 7 are the survived chromosomes to be added into a mating pool to generate the next generation chromosomes.

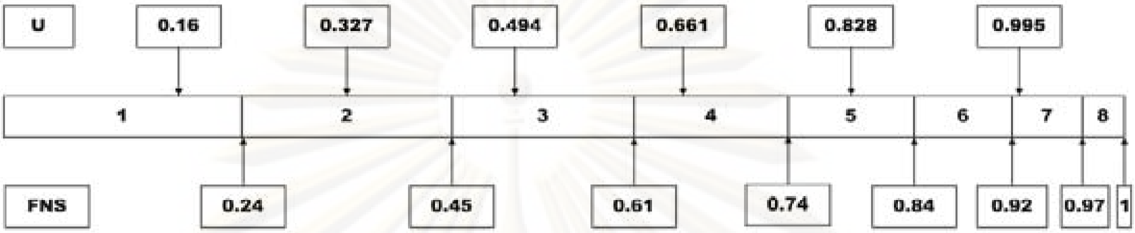

Figure 3.4: The SUS method.

## 3.4 IFGA Cross-Over

There are two problems in cross-over by the integer encoding method. The first problem is that some features cannot be in the same chromosome. The original cross-over of the genetic algorithm randomly selects the cross-over point and swaps two chromosomes from the cross-over point to the end of chromosomes. To do cross-over by the original genetic algorithm, features at the same loci cannot be in the same chromosome. Those loci may be the best values in the chromosome. To overcome this problem, a sequence of each chromosome was re-arranged before swapping values in chromosomes. The second problem occurs due to multiple sizes of the parent chromosomes. The original genetic algorithm allows only the equal size of chromosome. Therefore, the new cross-over process is proposed for the IFGA approach.

### IFGA Cross-over Algorithm

```
input:  1. parent$_1$ chromosome $x_1, x_2, ..., x_t$
        2. parent$_2$ chromosome $y_1, y_2, ..., y_u$
```

```
where
          1. $t, u$ are lengths of chromosomes $x$ and $y$, respectively.
          2. either $t > 1$ or $u > 1$.
          3. $t, u > 0$
output:   1. Two offspring chromosomes

begin
          Randomly select parents.
          Re-order values of each parent's chromosome.
          Randomly select a cross-over point, $c$,
           where $c \in [1, \min(t, u) - 1]$
          Swap two parent chromosomes to obtain the following
          result:
                    the first offspring $x_1, x_2, ..., x_c, y_{c+1}, \dots, y_t$ and
                    the second offspring $y_1, y_2, ..., y_c, x_{c+1}, \dots, x_u$.
end
```

From the algorithm, suppose that the cross-over point ($c$) is $2$ and two considered parents are $\text{parent}_1$: [73, 6, 25] and $\text{parent}_2$: [5, 11, 96, 36, 82]. Then, two offspring chromosomes are [73, 6, 96, 36, 82] and [5, 11, 25] after swapped, respectively.

Since we focus on the feature selection, it is obvious that the sequence of genes in the chromosome is not important. Each gene corresponds to each selected feature. For example, chromosome [73, 6, 25] is equivalent to chromosome [6, 25, 73] because both chromosomes contain the same set of genes or features.

## 3.5 IFGA Mutation

Mutation is a process of changing the value of a specified locus. It hardly occurs when compared with the cross-over process. In a binary encoding, if the chromosome is 01100, after the mutation occurs at the fifth position, this chromosome values is changed to 01101. However, mutation of a chromosome obtained by binary encoding cannot be applied to our integer encoding scheme. Suppose the defined number of selected feature is *M*. This implies that there are *M* positions in the sequence. Let a considered chromosome to be mutated have the following content [73, 6, 96, 36, 82]. If position 5 in this chromosome is selected as a mutation position then the number 82 at position 5 must be replaced by another number. This new number is randomly selected from the interval [1, M]. In this example, assume that the randomly selected number from [1, M] is number 3. Then, the mutated chromosome becomes [73, 6, 96, 36, 3].

IFGA Mutation Algorithm

```
input:  Chromosome $x = \{x_1, x_2, ..., x_n\}$ and length of genotype
        sequence.
output: A mutation chromosome.

begin
    randomly select a mutation point, $i$, $1 \leq i \leq n$
    randomly select a position number $f$ in the range $[1, M]$.
    replace $x_i$ with $f$.
end
```

## 3.6 Termination

This IFGA algorithm recursive steps of generating the population, evaluation by a BoostMode-SVM, IFGA selection, IFGA cross-over, and IFGA mutation until the number of the same best results remains the same to the next 300 iterations.

## 3.7 BoostMode-SVM

Prior to applying SVM, each number denoting the position of corresponding genotype in the considered sequence in any chromosome is substituting by its corresponding genotype. Each genotype is encoded in forms of 3 binary digits based on types of homozygote, homozygote, and heterozygote. In this paper, a new technique of oversampling for nominal feature is proposed to improve the performance of the SVM. The BoostMode-SVM generates 2 SVMs, $SVM_1$ and $SVM_2$. The $SVM_1$ is constructed for generating the score of the training data set. On the other hand, the $SVM_2$ is the final SVM model for classification the test set. First of all, all data are separated into training set and test set. Only the training set is used to construct the $SVM_1$. This training set is also used to find the BoostMode value, which is the indicator of data set. After the BoostMode is discovered, this BoostMode is brought to test with the $SVM_1$ model. Two scoring methods, an Unbiased Scoring (US) and a Bias Scoring (BS), are proposed. The US method is performed when the SVM correctly classifies the BoostMode data whereas the BS method is performed when the $SVM_1$ incorrectly classifies the BoostMode data. After scoring values are computed, a Scoring Over-Sampling approach (SOS) is processed to sample the data of the minority group until the number of data of both target groups is equal. Since this study classifies data set into 2 groups (+1 and -1), the minority group in this paper means the group of data set which has less elements. The new $SVM_2$ is constructed for the classification by the previous training data set and a new set of data from the SOS technique. Finally, the test set is run in the $SVM_2$ for the evaluation. The error rate for the test set is the *fitness_score* value using in the IFGA section above. The details about finding the BoostMode value, US, BS, and SOS approaches will be described in the following subsection.

### 3.8 Data Encoding

Since the given data are genotype SNPs which are characters but inputs to a classifier are numeric data, each individual genotype data must be encoded. Genotype data set is classified as a categorical variable. Each character of genotype is divided into three categories: major homozygote, minor homozygote, and heterozygote. Therefore, dummy encoding is applied for a SVM as vectors [1 0 0], [0 1 0], and [0 0 1] where a genotype is major homozygote, minor homozygote, and heterozygote, respectively.

| Genotype | Group | Encoded data for the BoostMode-SVM |
|---|---|---|
| CC TT AA | Case | 1 0 0 0 1 0 0 0 1 +1 |
| CT TG AT | Control | 0 0 1 0 0 1 0 0 1 -1 |
| TT TG TT | Case | 0 1 0 0 0 1 1 0 0 +1 |
| CT GG AT | Control | 0 0 1 1 0 0 0 0 1 -1 |
| TT TG AA | Case | 0 1 0 0 0 1 0 1 0 +1 |

Figure 3.5: The encoding technique for BoostMode-SVM.

Figure 3.5 shows an example of how the genotypes are encoded. Underline genotypes in this figure represent the major homozygote. The major homozygote and the minor homozygote are genotypes that have the same alleles, e.g. AA, TT, CC, or GG. The major homozygote is an allele that has more number of alleles than the minor homozygote. For instance, if a number of alleles A is 100 and the number of alleles T is 70, allele A is a major homozygote and allele T is a minor homozygote. On the other hand, the heterozygote is a genotype that has the different alleles such as CT, TG, AT. The last column of the encoded data is the target, where the case group is set to +1 and the control group is set to -1.

## 3.9 Adding Artificial Data to Minority Group

To make the data in both classes balanced, some additional data in minority group must be artificially generated. The selected generatng method (either US or BS) will depend upon a data called *BoostMode* data. This *BoostMode* data is used to test with the SVM. If it is correctly classified then the US method is used to generate new data. Otherwise the BS method is used instead. The following procedure describes how to compute BoostMode value and select a BoostMode data. Let $N_1$ be the number of data in the monority group. Boostrap sampling with replacement is applied on the minority group to generate $t$ data sets, i.e. $\{BoostGroup_1, \ldots, BoostGroup_t\}$. Each $BoostGroup_i$ set contains $N_1$ data.

### Finding BoostMode Data Algorithm

```
input:   $BoostGroup_1$, $BoostGroup_2$. …, $BoostGroup_t$
output:  BoostMode data.

begin
    for each $BoostGroup_i$ , $1 \leq i \leq t$ do
        Let $f_j^{(i)}$ be the frequency of occurrences of data $data_j^{(i)}$, for
        $1 \leq j \leq N_1$, in $BoostGroup_i$.
        Let $f_*^{(i)} = max_j(f_j^{(i)})$.
```

```
5.                  Let $data_*^{(i)}$ be the data whose frequency of occurrences is
                    $f_*^{(i)}$.
6.          end
7.          Assign a data in $G$ having maximum frequency of occurrence as
            BoostMode data.
8. end
```

To make the algorithm understandable, suppose the minority group contains the following five data. Three BoostGroups are generated by boostrap sampling with replacement.

Data in Minority group:

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

BoostGroup$_1$:

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

BoostGroup$_2$:

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

BoostGroup$_3$:

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

The following data are selected from each *BoostGroup$_i$* to create set *G:* [1 0 0 0 1 0 1 0 0 0 1 0], [1 0 0 0 1 0 1 0 0 0 1 0], [0 0 1 1 0 0 0 0 1 0 0 1]. From G, data [1 0 0 0 1 0 1 0 0 0 1 0] is selected as BoostMode data since it has highest frequency of occurrences.

### 3.10 The Unbiased Scoring Method

This technique is processed when the SVM$_1$ classifies the BoostMode correctly. All data points have equal chances (equal scoring values) to be selected for the over-sampling technique. The following algorithm describes the process of finding the scoring value by the US technique.

The Unbiased Scoring Algorithm

**input:** Set of $N_1$ data from the minority group.

**output:** Scoring value (*scoreVal*) of each data in the minority group.

```
begin
    for $i = 1 \text{ to } N_1$
        $\text{scoreVal}_i = \frac{1}{N_1}$
    end
end
```

### 3.11 The Bias Scoring Method

The BS technique is run when the $\text{SVM}_1$ incorrectly classifies by the BoostMode value. In this technique, each data point has its own probability to be selected for the sampling with replacement. The probability value is the scoring value which is calculated by the distance of its point to the decision hyperplane. Suppose group1 is the minority group. All training data points are classified to find the distance between itself and the decision hyperplane. This distance value will be applied to find the scoring value. The distance of each data point is calculated by the equation (3.2) for linear separability or the equation (3.3) for non-linear separability.

$$\text{distance}_i = \mathbf{w}^T \cdot \mathbf{x}_i + b \tag{3.2}$$

$$\text{distance}_i = \mathbf{w}^T \cdot \varphi(\mathbf{x}_i) + b \tag{3.3}$$

Since SVM concerns targets of -1 or +1 group, the distance value can be both positive and negative. In this paper, the target of -1 group is a group of control and the target of +1 group is a group of case. Assume that positive distance means the target of the data point is in group 1 and negative distance means the target of the data point is in group -1. The data point which is correctly classified has a less chance (less scoring value) to be selected for the over-sampling than the data point which is wrongly classified. Although, the data points which are in the same group, all data in the correctly classified group or incorrectly classified group, these data points also have different scoring value. The data points which are near the decision hyperplane will have more chances (more scoring value) to be selected than the data points which are far away from the decision hyperplane. Therefore, the more further with incorrect classification is, the more chances (more scoring value) to be sampled. The more further with correct classification the less chance (less scoring value) to be sampled. The scoring value for the BS method is illustrated by the following algorithm.

The Bias Scoring Value Algorithm

**input:** *distance* = {$distance_1$, $distance_2$, …, $distance_{N1}$}, of the minority group.

**output:** The scoring values (*scoreVal*) of the minority group.

```
begin
    $minVal = min(distance_1, …, distance_{N1})$
    $addVal = |minVal| + 1$
    for $i = 1$ to $N1$
        $scoreVal_i = distance_i + addVal$
    end
    for $i = 1$ to $N1$
        $scoreVal_i = \frac{\sum_{j=1}^{i} scoreVal_j}{\sum_{j=1}^{N1} scoreVal_i}$
    end
    if the minority group is the control group
        for $i = 1$ to $N1$
            $scoreVal_i = 2 - scoreVal_i$
        end
        for $i = 1$ to $N1$
            $\text{scoreVal}_i = \frac{\sum_{j=1}^{i} \text{scoreVal}_j}{\sum_{i=1}^{N1} \text{scoreVal}_i}$
        end
    end
end
```

### 3.12 The Scored Over-Sampling Method

After the scoring value is generated, the Scored Over-Sampling method (SOS) maps this score into a continuous value called *mapped_scoreVal* by equation (3.4). Once, the *scoreVal* is mapped, the SOS processes run as the following algorithm. All data in the minority group have the $scoreVal_i$ computed by either Bias Scoring Value Algorithm or Unbiased Scoring Value Algorithm. Let $MD_i$ denote data *i*, *for* $1 \leq i \leq N_1$, in the minority group. The number of data in minority group and majority groups are $N_1$ and $N_2$, respectively.

$$\text{mapped\_scoreVal}_1 = \left[ 0, \frac{\text{scoreVal}_1}{\sum_{j=1}^{N1} \text{scoreVal}_i} \right]$$

(3.4)

$$\text{mapped\_scoreVal}_1 = \left[ \frac{\sum_{j=1}^{j-1} \text{scoreVal}_i}{\sum_{j=1}^{N1} \text{scoreVal}_i}, \frac{\sum_{j=1}^{j} \text{scoreVal}_i}{\sum_{j=1}^{N1} \text{scoreVal}_i} \right] \text{where } j > 1$$

The SOS Algorithm

**input:** 1. The minority data set *MD* to be sampled.

2. Number of data to be added in this sample, $N_2$-$N_1$.

3. The Mapped scoreVal, *mapped_scoreVal*.

**output:** 1. The sampled data, *samp_data*.

```
begin
    for $i = 1$ to $N_2 - N_1$
        $selectPosition = rand(1)$
        for $j = 1$ to $N_2 - N_1$
            if the selectPosition value in the range of
                $mapped\_scoreVal_j$
                $samp\_data_i = MD_j$
            end
        end
    end
end
```

# CHAPTER 4

# EXPERIMENTAL RESULTS

## 4.1 Data Sets

Two data sets from Crohn's disease and beta$^0$/Hb E. Thalassemia disease are tested. Crohn's disease data set for case-control study **[21]** are sequenced from 616 kilobase at chromosome 5q31. There are 103 genotype SNPs which consist of 243 controls and 114 cases. To compare with an optimum random forest (ORF) **[16]**, missing data have to inferred by 2SNP phasing method **[22]**. Beta$^0$/Hb E Thalassemia disease data set are obtained from whole genome of all regions in Thailand. The SNP genotype data sets were given from Thalassemia Research Center, Mahidol University, Thailand. There are entire beta$^0$/Hb E Thalassemia patients. These data consist of 198 controls and 305 cases. Each subject consists of 835 genotype SNPs.

## 4.2 Parameters Tuning

The parameters were tuned to find the suitable values for the IFGA with BoostMode-SVM approach. This part represents a pilot study for tuning the number of replicated bootstrap groups for the BoostMode-SVM technique and the cross-over and mutation rates for the IFGA technique.

### 4.2.1 A Number of Replicated Bootstrap

In the theory of the bootstrap technique, the number of replicated bootstrap groups has to be defined. The number of bootstrap replications was varied from 1 to 100,000 to find the stable mode value. It can be seen that the mode value of Crohn's disease (Figure 4.1) does not change in any number of replications whereas the mode value of Thalassemia (Figure 4.2) is not stable when the number of replications is less than 1700.

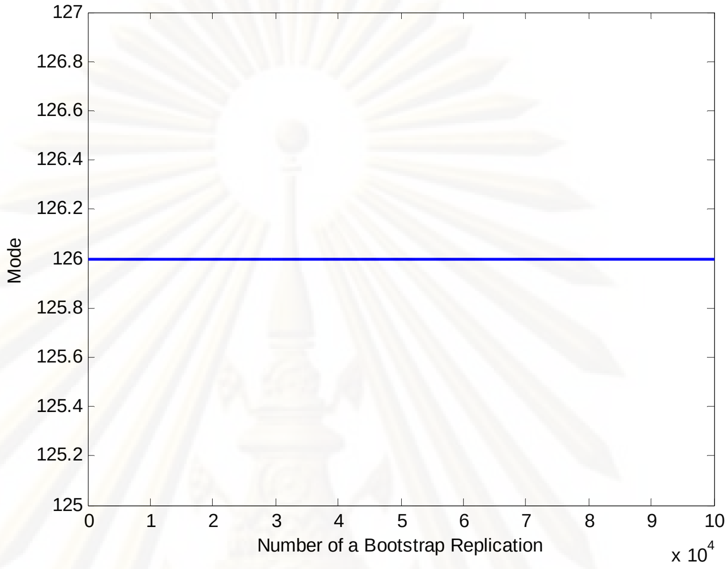

Figure 4.1: The number of bootstrap replication of Crohn's disease.

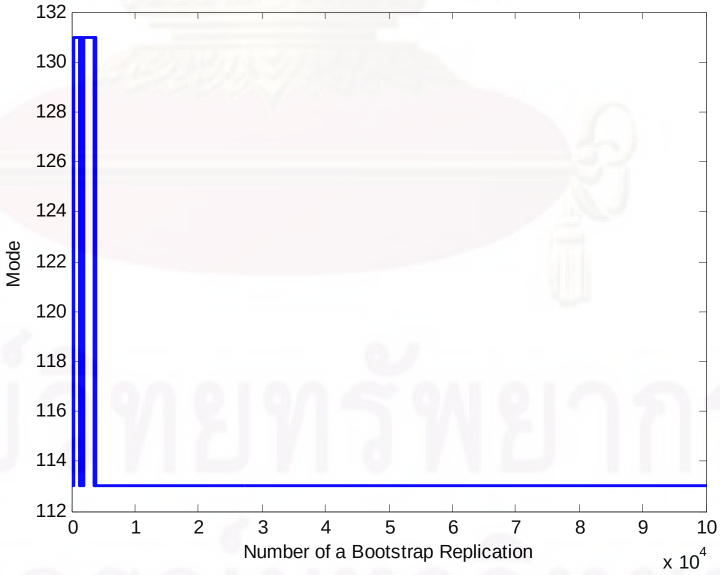

Figure 4.2: The number of bootstrap replication of Thalassemia's disease.

### 4.2.2 A Cross-Over Rate Tuning

The cross-over rate is varied from 0.5 to 0.95 for both Crohn's and Thalassemia's diseases with 1000 population. The cross-over rate is set to 0.8 for Crohn's disease (Figure 4.3) and 0.7 for Thalassemia's disease (Figure 4.4), respectively, due to the minimum value of fitness score.

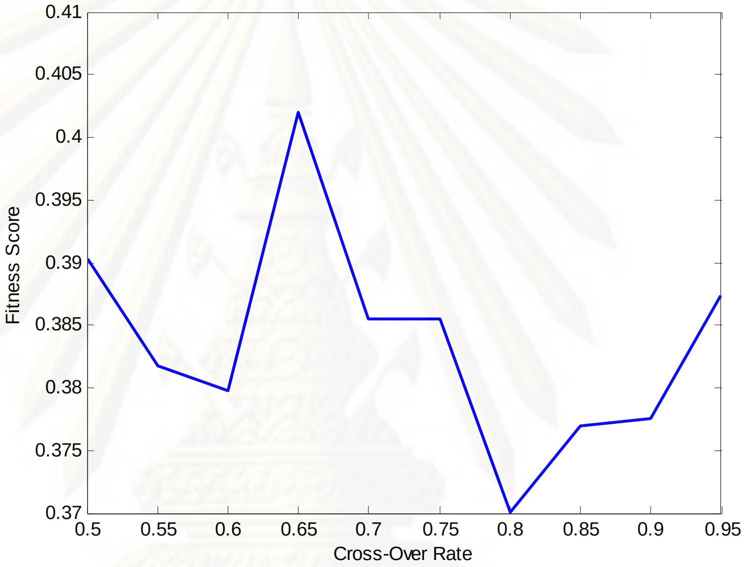

Figure 4.3: Cross-Over Rate of Crohn's disease.

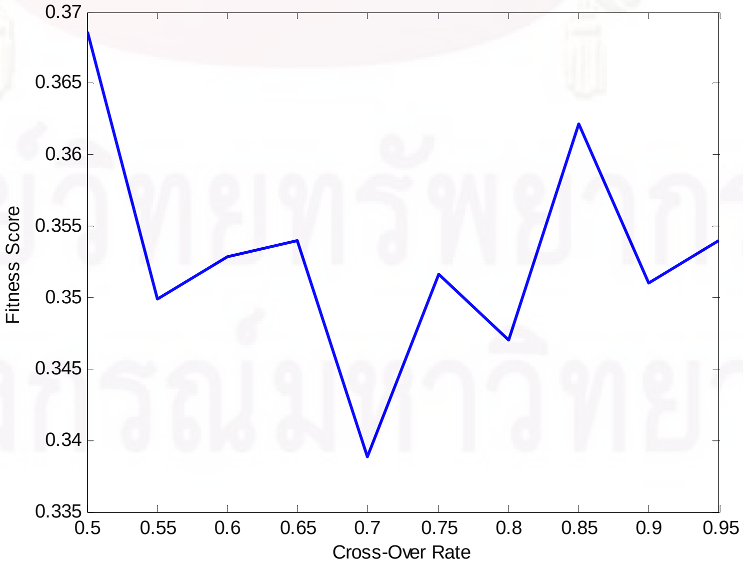

Figure 4.4: Cross-Over Rate of Thalassemia's disease.

### 4.2.3 A Mutation Rate Tuning

The following graphs represent tuning a parameter of mutation rate of Chohn and Thalassemia diseases. The mutation rate is varied from 0.001 to 0.05 for both Crohn's and Thalassemia's diseases with 1000 population. The mutation rate is set to 0.001 for Crohn's disease (Figure 4.5) and 0.035 for Thalassemia's disease (Figure 4.6), respectively, due to the minimum value of fitness score.

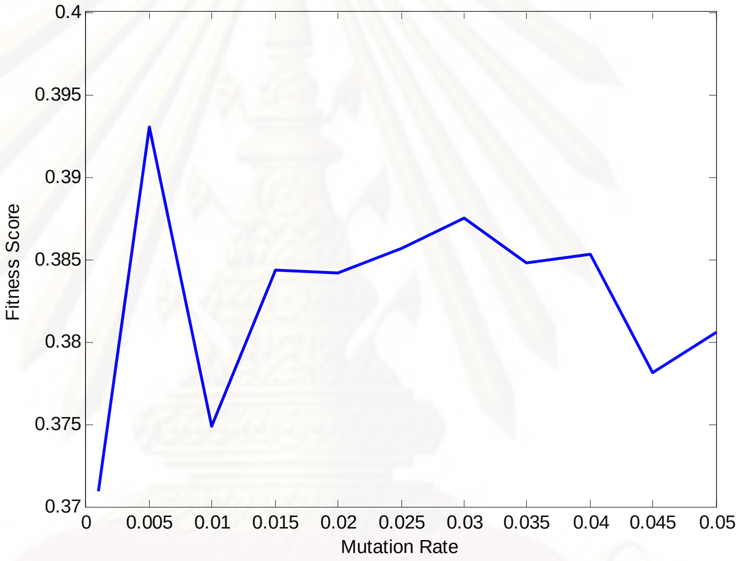

Figure 4.5: Mutation Rate of Crohn's disease.

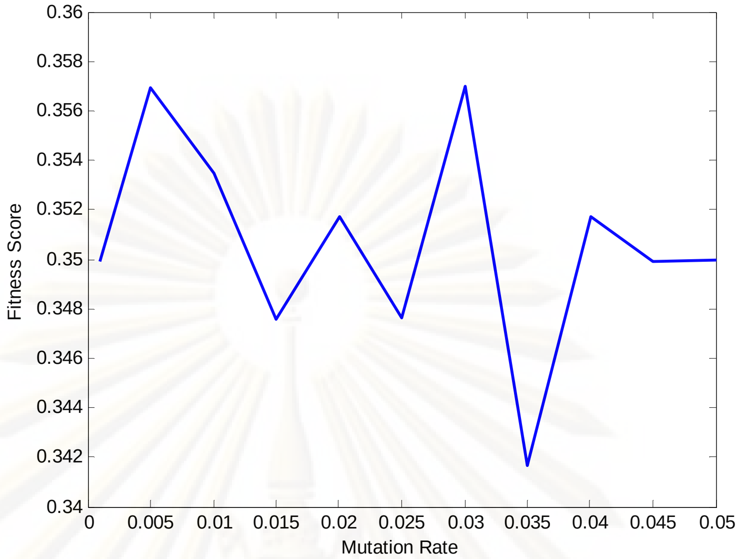

Figure 4.6: Mutation Rate of Thalassemia's disease.

### 4.3 Performance Measures

Table 4.1 and Table 4.2 show a comparison of the IFGA-BoostMode-SVM, ORF, and CART by 10-fold cross validation of Thalassemia's and Crohn's diseases respectively. Our IFGA with the BoostMode-SVM method performs better classification than the ORF method.

Table 4.1: The experimental results of Thalassemia's disease.

| Algorithm | # feature | accuracy | sensitivity | specificity |
|---|---|---|---|---|
| IFGA-BoostMode-SVM | 6 | 71.57 | 76.39 | 64.14 |
| ORF | 6 | 54.27 | 69.84 | 30.30 |
| CART | 6 | 69.38 | 76.07 | 59.09 |

Table 4.2: The experimental results of Crohn's disease.

| Algorithm | # feature | accuracy | sensitivity | specificity |
|---|---|---|---|---|
| IFGA-BoostMode-SVM | 8 | 71.06 | 62.50 | 76.13 |
| ORF | 8 | 57.88 | 20.14 | 80.25 |
| CART | 8 | 63.31 | 23.61 | 86.83 |

Table 4.3 represents a comparison of the IFGA-BoostMode-SVM from all features and selected features for Thalassemia’s and Crohn’s diseases. The result shows that using all features gives less accuracy than using only selected features.

Table 4.3: The experimental results from all features versus selected features of Thalassemia's and Crohn’s diseases.

| Data Set | # Feature | Accuracy |
|---|---|---|
| Thalassemia | All | 60.64 |
| | 6 | 71.57 |
| Crohn | All | 64.60 |
| | 8 | 71.06 |

Feature selection in the IFGA approach uses the principle of survival. Chromosome producing more accuracy has more chance for survival and generating offspring. Table 4.5 and Table 4.6 present best of the best chromosome from all generations of Thalassemia’s and Crohn’s diseases by the IFGA with BoostMode-SVM, respectively. It has been noted that, there is no generation that has the best chromosome containing only 1 feature for Thalassemia’s disease. For Crohn’s disease, there is no generation that has the best chromosome containing 1, 3, 4, and features.

Table 4.4: The experimental results of best of the best chromosome from all generations of Thalassemia's disease.

| # feature | accuracy | sensitivity | specificity |
|---|---|---|---|
| 1 | - | - | - |
| 2 | 58.45 | 64.26 | 49.49 |
| 3 | 65.81 | 81.97 | 40.91 |
| 4 | 67.59 | 80.00 | 48.48 |
| 5 | 70.58 | 77.38 | 60.10 |
| 6 | 71.57 | 76.39 | 64.14 |
| 7 | 71.37 | 80.00 | 58.08 |
| 8 | 67.59 | 63.28 | 74.24 |
| 9 | 67.00 | 62.62 | 73.74 |
| 10 | 62.82 | 90.16 | 20.71 |

Table 4.5: The experimental results of best of the best chromosome from all generations of Crohn's disease.

| # feature | accuracy | sensitivity | specificity |
|---|---|---|---|
| 1 | - | - | - |
| 2 | 63.57 | 36.81 | 79.42 |
| 3 | - | - | - |
| 4 | - | - | - |
| 5 | - | - | - |
| 6 | 64.34 | 56.25 | 69.14 |
| 7 | 6641 | 56.94 | 72.02 |
| 8 | 71.06 | 62.50 | 76.13 |
| 9 | 69.77 | 65.28 | 72.43 |
| 10 | 68.73 | 52.08 | 78.60 |

For a SVM, a soft margin RBF kernel function with sigma = 0.5 is deployed in both Crohn's dataset and Thalasemia's dataset.

In IFGA, the first population is generated by choosing an arbitrary number of features as a number of susceptible loci. Then, feature selection with less than or equal to the number of defined features are processed. Parameters in GA were set as follows:

1. the number of chromosomes is set to 1000;
2. the stopping criteria is when the best answer remains the same in the next 300 generations;
3. the fitness function is an error of classification;
4. the cross-over rate is 0.7 for Thalassemia's and 0.8 for Crohn's diseases, respectively;
5. the mutation rate is 0.035 for Thalassemia's and 0.001 for Crohn's diseases, respectively.

The accuracy, the sensitivity, and the specificity are calculated as following equations:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{4.2}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{4.3}$$

If the predicted value is more than the threshold and the desired output class is case then the prediction is true and true positive (TP) is accounted. On the contrary, if the prediction value is less than the threshold then the predicted is false and the false negative (FN) occurs. In case of true negative (TN), if the predicted value is less than the threshold and the desired output class is control then the prediction is true. Conversely, if the prediction value is greater than the threshold then the prediction is false and the false positive (FP) occurs. Figure 4.7 and Figure 4.8 demonstrate the fitness score of Chohn's and Thalassemia's diseases.

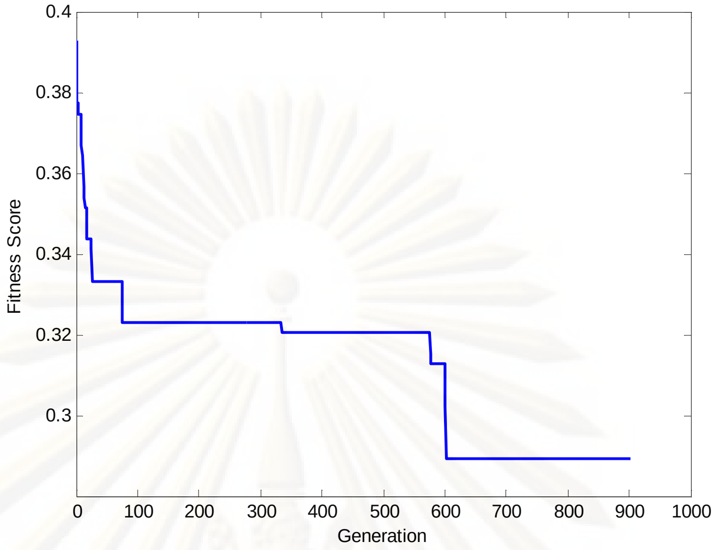

Figure 4.7:  Fitness score of Crohn's disease.

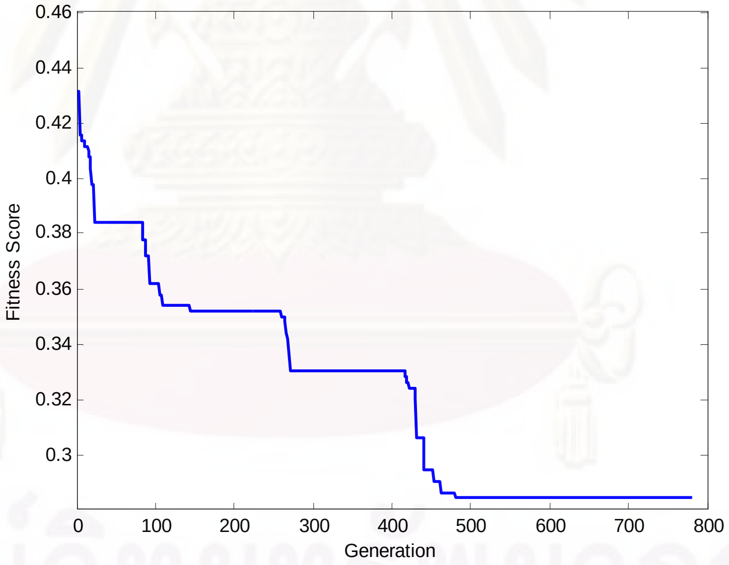

Figure 4.8:  Fitness score of Thalassemia's disease.

Two ROC curves show a comparison between IFGA with the BoostMode-SVM and ORF technique of Crohn's disease (Figure 4.9) and Thalassemia's disease (Figure 4.10). From both figures, IFGA with the BoostMode-SVM clearly outperforms to the ORF classifier. A strange line is the ROC curve of the IFGA with BoostMode-SVM technique and the dash line is the ROC curve of the ORF technique.

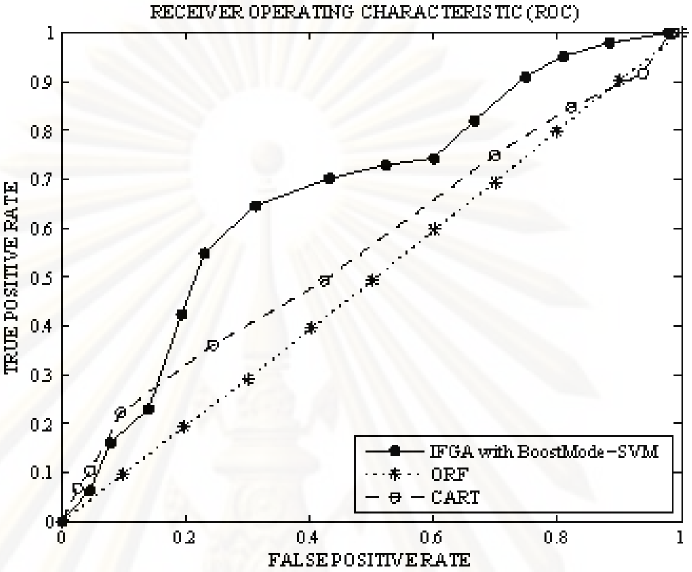

Figure 4.9: The ROC curve of Crohn's disease.

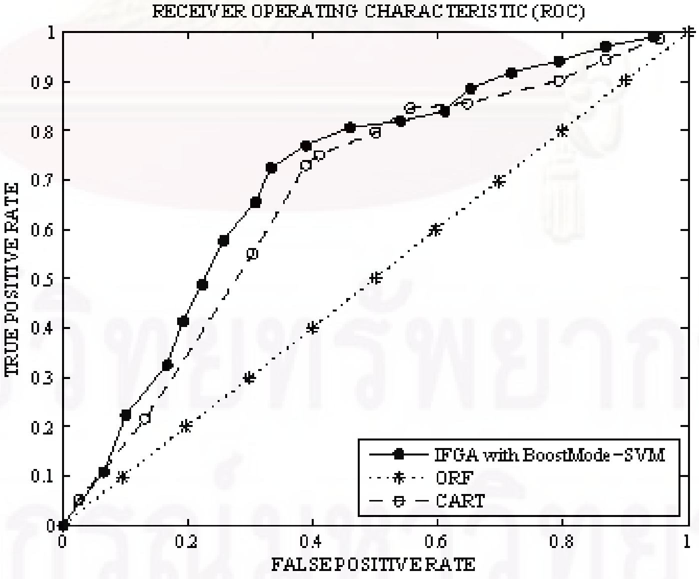

Figure 4.10: The ROC curve of Thalassemia's disease.

# CHAPTER 5

# CONCLUSION

Identifying the susceptibility loci is a method to discover an association of a particular disease. This dissertation proposes two main algorithms to find the susceptibility loci of a case-control association: (1) the new genetic algorithm approach for selecting the informative SNPs (IFGA), and (2) the new SVM approach for classifying imbalance data set (BoostMode-SVM). The utilization of the proposed IFGA with BoostMode-SVM is to select features or loci of SNPs that give high accuracy for classification of the case-control data. The value of each chromosome represents the positions of genotype sequence. Chromosomes are encoded as an integer with variable sizes. The length of chromosome does not have equal number of all input features. This technique reduces running time when only few selected features are chosen from the huge length of the chromosome. The original cross-over and mutation techniques have to be adapted. After the chromosomes are generated, those chromosomes will be evaluated by the BoostMode-SVM.

The BoostMode-SVM is a new SVM classifier approach. It generates new artificial data points of the minority group by a scored-over sampling approach called SOS. The SOS technique over samples the minority group by two scoring approaches: BS and US. The BoostMode-SVM applied the Bootstrap technique to select the value of BoostMode. This BoostMode is the indicator to tell the program which bias or unbiased scoring technique is applied. The BoostMode-SVM constructs two SVM models. The first model is used for classifying the BoostMode value. If the BoostMode value correctly classifies the data, the US approach is processed. On the other hand, if the BoostMode value incorrectly classifies the data, the BS approach is processed.

The results of two real data sets: Crohn's and Thalassemia's diseases show that using all features to classification case-control data does not give the highest accuracy. Feature selection and classification by the IFGA with BoostMode-SVM clearly outperforms the ORF technique. The IFGA with BoostMode-SVM with 6 loci gave 71.57%

correction for Thalassemia's disease with 76.39% sensitivity and 64.14% specificity. However, the accuracy, sensitivity, and specificity of Crohn's disease with 8 loci are 71.06%, 62.50%, 76.13%, respectively.

Finally, based on the data of Thai patients, SULF1 (chromosome 8), unknown (chromosome 9), NP_001002265.1 (chromosome 10), unknown (chromosome 11), KIAA0769 (chromosome 11), ME3 (chromosome 11) are the candidates for Thalassemia's biomarkers.

# References

[1] F. Pardi, F., Lewis, C. M. and Whittaker, J. C. SNP Selection for Association Studies: Maximizing Power across SNP Choice and Study Size. Annals of Human Genetics 69(2005) : 733-746.

[2] Marchini, J., Donnelly, P. and Cardon, L. R. Genome-Wide Strategies for Detecting Multiple Loci that Influence Complex Diseases. Nature Genetics 37(2005) : 413-417.

[3] Weatherall, D. J. Science, Medicine, and the Future: Single Gene Disorders or Complex Traits: Lessons from the Thalassaemias and other Monogenic Diseases. BMJ 321(2000) : 1117-1120.

[4] Libioulle, C., et al. Novel Crohn Disease Locus Identified by Genome-Wide Association Maps to a Gene Desert on 5p13.1 and Modulates Expression of PTGER4. PLoS Genet 3(2007) : 538-543.

[5] Chatterjee, N., Kalaylioglu, Z., Moslehi, R., Peters, U. and Wacholder, S. Powerful Multilocus Tests of Genetic Association in the Presence of Gene-Gene and Gene-Environment Interactions. Am J Hum Genet 79(2006) : 1002-1016.

[6] Rockenbauer, E., et al. SNP Genotyping Using Microsphere-Linked PNA and Flow Cyto-metric Detection. Cytometry Part A 64A(2005) : 80-86.

[7] Shah, S. C. and Kusiak, A. Data mining and genetic algorithm based gene/SNP selection. Artificial Intelligence in Medicine 31(2004) : 183-196.

[8] Saeys, Y., Inza, I., and Larrañaga, P. A Review of Feature Selection Techniques in Bioinformatics. Bioinformatics 23(2007) : 2507-2517.

[9] Lai, C., Reinders, M. JT., van't Veer, L. J., and Wessela, L. FA. A Comparison of Univariate and Multivariate Gene Selection Techniques for Classification of Cancer Datasets. BMC Bioinformatics 7(2006) : 235.

[10] Gerger, A., et al. A mulitgenic approach to predict breast cancer risk. Epidemiology 104(2007) : 159-164.

[11] Cordell, H. J., and Clayton, G. D. Genetic association studies. The Lancet 366(2005) : 1121-1131.

[12] Chen, J. S., and Hou, J. L. A Combination Genetic Algorithm with Application on Port-folio Optimization. M. Ali and R. Dapoigny 4031(2006) : 197-206.

[13] Cortes, C., and Vapnik, V. Support-Vector Networks. Machine Learning (1995) : 273-297.

[14] Kerstann, K. F., et al. Identification of susceptibility loci for complex diseases in a case-control association study using the Genetic Analysis Workshop 14 dataset. BMC Genetics 6(2005) : S102.

[15] Ritchie, M. D., et al. Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. Am. J. Hum. Genet 69(2001) : 138-147.

[16] Mao, W., and Kelly, S. An Optimum Random Forest Model for Prediction of Genetic Susceptibility to Complex Diseases. LNAI 4426(2007) : 193-204.

[17] Zeng, X. P., Yong-Ming Li, Y. M., and Qin, J. A Dynamic Chain-Like Agent genetic Algorithm for Global Numerical Optimization and Feature Selection. Neurocomputing 72(2009) : 1214-1228.

[18] Baker, J. E. Reducing Bias and Inefficiency in the Selection Algorithm. Proceedings of the Second International Conference on Genetic Algorithm and their Application (1987) : 14-21.

[19] Majumdar, J., and Bhunia, A. K. Elitist Genetic Algorithm for Assignment Problem with Imprecise Goal. European Journal of Operational Research 177(2007) : 684-692.

[20] ChiLam, K. C., Ning, X., and Gao, H. The Fuzzy GA-Based Multi-Objective Financial Decision Support Model for Chinese State-Owned Construction Firms. Automation in Construction 18(2009) : 402-414.

[21] Daly, M., Rioux, J., Schaffner, S., Hudson, T., and Lander, E. High Resolution Haplotype Structure in the Human Genome. Nature Genetics 29(2001) : 229-232.

[22] Brinza, D., and Zelikovsky, A. 2SNP : Scalable Phasing Method for Trios and Unrelated Individuals. Journal of IEEE/ACM Transactions on Computational Biology and Bioinformatics 5(2008) : 313-318.

# Biography

**Name:** Miss Khantharat ANEKBOON.

**Date of Birth:** 9th July, 1979.

**Educations:**

Ph.D. Candidate in Computer Science, Department of Mathematics, Chulalongkorn University, Thailand, (October 2004 - September 2009).

M.Sc. Program in Information Technology (International Programme), Faculty of Information Technology, King Mongut's Institute of Technology North Bangkok, Thailand (May 2000 - May 2004).

B.Sc. Program in Computer Science, Faculty of Science, Thammasat University, Thailand (June 1996 - February 2000).

**Scholarship:** The Office of Higher Education Commission, Ministry of Education, Thailand.