FEATURE SELECTION
USING
GENETIC ALGORITHMS

by

SEVİM EDA BARLAK

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of the
requirements for the degree of
Master of Science

SABANCI UNIVERSITY
Summer 2007

FEATURE SELECTION
USING
GENETIC ALGORITHMS

APPROVED BY:

Assoc. Prof. Uğur Sezerman (Thesis Supervisor) ………………..

Assoc. Prof. Hikmet Budak ………………..

Assoc. Prof. Batu Erman ………………..

Assist. Prof. Devrim Gözüaçık ………………..

Assist. Prof. Kemal Kılıç ………………..

DATE OF APPROVAL: 14.08.2007

ABSTRACT

Microarray data is very important for identification of complex diseases and the development of diagnostic kits. This topic exhibits considerable aid especially to cancer research. Therefore, an influential number of biological and medical researchers have to deal with the datasets obtained from microarray experiments. Usage of these huge datasets is not efficient in terms of time and cost. Thus, many researchers contribute to tumor classification via effective use of microarray technologies for cancer research. To be able to obtain the most relevant subset containing the signature genes that are included in the pathway of certain diseases and therefore capable of classifying the entire data, is very crucial for true disease diagnosis.

There are several approaches in the literature for this classification purpose. In this thesis, we present an approach to use, Genetic Algorithms for this feature subset selection problem. Genetic Algorithm is combined with Support Vector Machines for the calculation of classification accuracies of each gene. These classification accuracies denote the survival probabilities of the genes in our algorithm. The genes having higher classification accuracy will have more probability to survive.

Three different real life cancer datasets are used for the tests. Our algorithm converged to better results then all other approaches in the literature. In colon tumor dataset which is one of our test datasets, we were able to classify the entire data with the accuracy of 100% using only 4 features (genes). In prostate cancer dataset we classified the data using 3 features with the accuracy of 100%. And finally we tested our Genetic Algorithm using an ovarian cancer dataset and we found only 3 significant features out of 15154 genes, again with the accuracy of 100%.

ÖZET

Günümüzde hastalıkların tespit edilmesinde ve tanı aparatlarının geliştirilmesinde microarray verilerinin çok önemli bir yeri vardır. Bu konu özellikle kanser araştırmalarında büyük rol oynamaktadır. Bu sebeple, biyolojik alanda ve tıp alanında çalışan çok sayıda araştırmacı microarray veritabanlarından elde edilmiş verilerle çalışmak zorundadır. Fakat bu veritabanları çok büyük miktarda veri içermektedir, dolayısı ile zaman ve bununla bağlantılı olarak maliyet açısından verimli çalışmak mümkün değildir. Bu sebeple, çok sayıda araştırmacı bu kanser veritabanlarını anlamlı şekilde ayıklamaya katkıda bulunma çalışmaları yapmaktadır. Bunun sebebi, kanser verisini ayırdetmeye yarıyacak ve tanıda kolaylık sağlayacak kayda değer genlerden oluşan bir alt küme bulmak, doğru teşhis açısından büyük önem taşır.

Verileri ayırdetme konusunda literatürde bir çok farklı yaklaşım mevcuttur. Bu tezde, bu *özellik altkümesini seçme* problemine bizim sunduğumuz yaklaşım *Genetik Algoritmalar* kullanmaktır. Bu altkümeyi oluşturabilecek özelliğin (genin) veriyi doğru sınıflandırma becerisini hesaplayabilmek için Genetik Algoritma ile birlikte *Destek Vektör Makinaları* kullanıldı. Bu sınıflandırma becerileri aynı zamanda her genin hayatta kalabilme olasılığını hesaplamaktadır. Yüksek sınıflandırma becerisine sahip genler, diğer genlere oranla daha yüksek olasılıkla kullanılacak ve algoritmanın ileri safhalarında da rol alma şansına sahip olacaklardır.

Algoritmanın testlerinde, gerçek hayattan elde edilmiş kanser verilerinden oluşan üç farklı kanser veritabanı kullanılmıştır. Testler sonucunda, sunmuş olduğumuz bu yaklaşımla literatürde bugüne kadar elde edilmiş sonuçlardan çok daha iyi sonuçlar elde etmiş olduğumuzu gördük. Kullanmış olduğumuz veritabanlarından biri kolon kanseri verileri içermektedir. Testler sonucunda yalnızca 4 gen kullanarak bütün veriyi %100 doğru olarak ayırdedebildiğimizi gördük. Bir diğer veritabanından elde ettiğimiz prostate kanseri verisini sadece 3 gen kullanarak yine %100 doğru ayırdedebilmeyi başardık. Son olarak Genetik Algoritmamızı bir ovaryen veritabanı üzerinde denedik ve her birey için 15154 genden oluşan bu veritabanını da yalnız 3 gen kullanarak %100 doğru olarak ayıredebildik.

ACKNOWLEDGEMENT

I would like to thank my supervisor Assoc. Prof. O. Uğur Sezerman for encouraging and supporting me in every phase of this work. Without his guidance this thesis would never come out.

I would like to express my gratefulness to my professors who were the members of my thesis jury, Hikmet Budak, Batu Erman, Devrim Gözüaçık and Kemal Kılıç for their leading advises.

I wish to thank Galip Gürkan Yardımcı, Alper Küçükural and all my friends in the Biological Sciences and Bioengineering Department and Materials Sciences and Engineering Department for their friendship and aid through my past years in Sabancı University.

I also wish to express my hearty thanks to my fiancé Can Devrim, for his incomparable support and belief that sustained and energized me in my hard times through this thesis.

Finally, I wish to express my thanks to my family. I would especially like to thank my mother for supporting me with love and self-denial in this period of my life, from the beginning to the end, like she did in every part of my life. It is a pleasure to dedicate this work to my sisters, my mother and the memory of my father.

TABLE OF CONTENTS

ABSTRACT v

ÖZET vi

ACKNOWLEDGEMENT vii

LIST OF FIGURES x

LIST OF TABLES xi

ABREVIATIONS xii

1 INTRODUCTION 1

1.1 Organization Of Thesis 2

2 OVERVIEW 3

2.1 Microarray 3

2.1.1 Gene Expression 3

2.1.2 Microarray Experiments 7

2.1.3 Different Microarray Technologies 9

2.1.4 The Usage Of Microarrays 9

2.2 Feature Selection 11

2.3 Genetic Algorithms 13

2.3.1 Encoding 15

2.3.2 Initialization of the Population 17

2.3.3 Fitness Function 17

2.3.4 Cross-over 18

2.3.5 Mutation 19

3 METHODOLOGY 21

3.1 About the Algorithm 21

3.1.1 Data Structure 21

3.1.2 Random Number Generator 22

3.1.3 Calculation of the Fitness 22

3.2 The Flow of the GA 23

3.2.1 Runs for Searching the Fitness Space 25

3.2.2 The Main Algorithm 28

4 RESULTS 34

4.1 Datasets 34

4.1.1 Previous Approaches Using The Same Datasets 35

4.2 Experiments 36

4.2.1 Experiments with Five-Fold Cross-Validation 37

4.2.2 An Experiment with Leave-One-Out Cross-Validation 40

4.3 Independent Tests to Verify Our Results 41

5 CONCLUSION AND DISCUSSION 42

5.1 Biological Relevancy of Our Results 45

6 REFERENCES 47

LIST OF FIGURES

Figure 2.1 Example of a Nucleotide 4

Figure 2.2 Base-paring 4

Figure 2.3 DNA, Helix Structure 5

Figure 2.4 Gene Expression 7

Figure 2.5 A Microarray Experiment 8

Figure 2.6 Crossing-over Chromosomes 14

Figure 2.9 One-Point Cross-over 18

Figure 2.11 Cut and Splice Cross-over 18

Figure 2.12 Uniform Mutation 20

Figure 3.1 The margin between two classes achieved from SVM 23

Figure 3.2 Cross-over in the Runs for Searching the Fitness Space 27

Figure 3.3 Calculating the Fitness Values of the Genes 28

Figure 3.4 Comparison of Some Cross-Over Techniques 30

Figure 3.4 A Match Causing No Shrink 32

Figure 3.5 A Match Causing a Shrink 32

Figure 4.1 The pattern of the results 40

LIST OF TABLES

Table 4.1 Classification Accuracy of the Results from Colon Cancer Dataset Compared with the Other Results in the Literature 37

Table 4.2 Classification Accuracy of the Results from Prostate Cancer Dataset Compared with the Other Results in the Literature 38

Table 4.3 Classification Accuracy of the Results from Ovarian Cancer Dataset Compared with the Other Results in the Literature 39

Table 5.1 Genes Achieved from Colon Tumor Dataset……………………………………….45

Table 5.2 Genes Achieved from Colon Tumor Dataset……………………………………….46

ABREVIATIONS

A	Adenine
C	Cytosine
cDNA	Complementary Dioxyribonucleic Acid
CFS	Correlation-Based Feature Selection
Cy 3	Cyanine 3
Cy 5	Cyanine 5
DNA	Deoxyribonucleic Acid
G	Guanine
GA	Genetic Algorithms
HPN	Hepsin (Transmembrane Protease, Serine 1)
HSP	Heat Shock Protein
***k*–NN**	*k* –nearest neighbor algorithm
LOOCV	Leave-One-Out Cross Validation
MAP	mitogen-activated protein
MAPKAPK3	Mitogen-Activated Protein Kinase-Activated Protein Kinase3
mRNA	Messenger Ribonucleic Acid
MT	Mersenne Twister
Mts-1	A Metastasis-Related Gene

NB	Naïve Bayes
NMHC	Myosin Heavy Chain, Non-Muscle
PCA	Principle Component Analysis
PCL	Prediction by Collective Likelihood of emerging patterns
RNA	Ribonucleic Acid
SEMGII	Semenogelin II
SVM	Support Vector Machines
STRAP	Cell Serine/Theorine Kinase Receptor
T	Thymine
TRIP 1	Homosapiens Thyroid Receptor Interactor/ TGβ protein
tRNA	Transfer Ribonucleic Acid
TβRII	The type II TGFβ
U	Uracil

1 INTRODUCTION

Genomics refers to have a wide scope of study of genes and their function. Progresses in bioinformatics such as microarray data analysis are very important for understanding the gene regulation mechanisms. Microarray experiments provide us with measuring the expression levels of thousands or ten thousands of genes in a single experiment and this technology allows examining gene expression patterns that have distinctive qualities for certain diseases. In today's world, a crucial number of biological and medical experts have been contributing to tumor classification via use of microarray technologies for cancer research.

Microarray experiments evincing significant issues in terms of diagnostics of relevant diseases generally bring about large datasets in which there are thousands of genes. Due to the fact that biological and medical researchers might suffer from difficulties of the use of such large datasets, this huge amount of data should be filtered. A possible solution to this problem is to obtain considerable set of genes which indicates the diseased individuals, via comparing the disease and the control data sets [1]. This subset consisting of significant genes may be useful for developing diagnostic kits and least number of genes would require least cost of test, would be more time efficient. Because of this reason, although reaching the exact classification in such large datasets using the least number of genes is not an easy responsibility, it is extremely important for successful diagnosis and treatment.

According to the facts discussed at the previous paragraph, the importance of determining the subset consisting of minimum number of genes which enables the classification of the disease data with the highest accuracy can not be disregarded. The problem might be associated with the selection of useful set of attributes rather than mutually redundant and irrelevant ones. In addition to the fact that this abundance of the ineffectual attributes causes a needless increase in the work space, it may also decrease the accuracy of the classification algorithm. The feature selection, which is also known as the subset selection, plays an important role in classification. Feature selection methods are frequently used in machine learning. In this process a subset of those features available from the data are selected, while the unimportant features are not taken into consideration. The best subset comprehends the least number of attributes, which we hope to aid to reach the highest accuracy.

Our proposition for minimizing the number of genes in a large dataset, is to treat the dataset with Genetic Algorithms using roulette wheel selection. Support Vector Machines are used to analyze the classification accuracy (the character) of the genes.

1.1 Organization Of Thesis

Chapter 2 is a review part in which describes the microarray technology in detail. The biological and the historical background of our approach are also mentioned in this chapter. The methodology used in this thesis is covered in Chapter 3. Results and the various experiments done using three different datasets are included in the following chapter; Chapter 4. The last chapter is the conclusion and the discussion part of the work.

2 OVERVIEW

2.1 Microarray

DNA microarray technology has the ability to analyze thousands of genes at the same time. This simultaneous analysis acquired by the ability of microarray technology to measure the activities and interactions of the related genes. In DNA microarray the method used is mostly based on the comparisons of the expression levels of the genes. So that to be able to understand uses of microarray technology a simple knowledge of the elementary mechanism of gene expression is necessary.

2.1.1 Gene Expression

Having knowledge of deoxyribonucleic acid (DNA) and ribonucleic acid (RNA) is crucial to understand gene expression. Both DNA and RNA consist of sequences of lined nucleotides. The nucleotides consist of a base, a sugar and a phosphate group (Figure 2.1). The sugar phosphate backbone (sugar and phosphate group) are bound to one after another by phosphodiester bond, in a row forming the linear strand posture of the DNA and the RNA. On the other hand the bases protrude from the sugars.

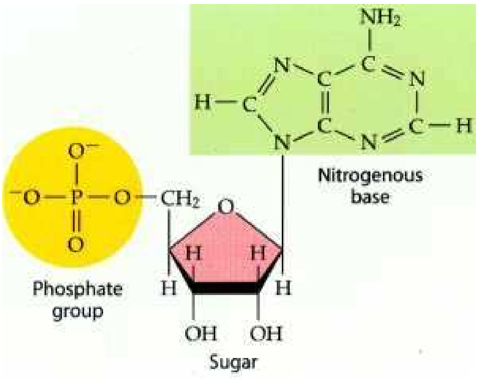


Figure 2.1 Example of a Nucleotide [16]

The bases that are mentioned above can be paired with each other via hydrogen bonds. An adenine (A) pairs with a thymine (T) and a guanine (G) pairs with cytosine (C) since they complement with each other in a way to maximize the number of hydrogen bonds (Figure 2.2).

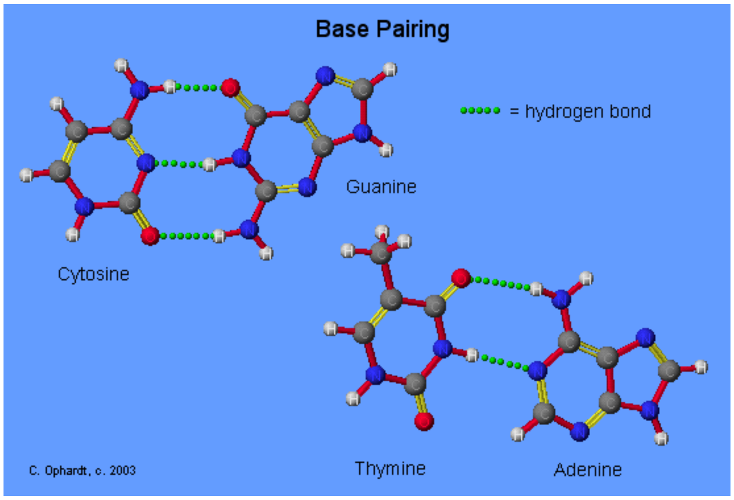


Figure 2.2 Base-paring [17]

If this base-pairing occurs between two complementary strands, this form of the DNA is called double-stranded which is also known with its three-dimensional helix structure as shown in the Figure2.3. According to the principle of base pairing, the complementary DNA (cDNA) assembles.

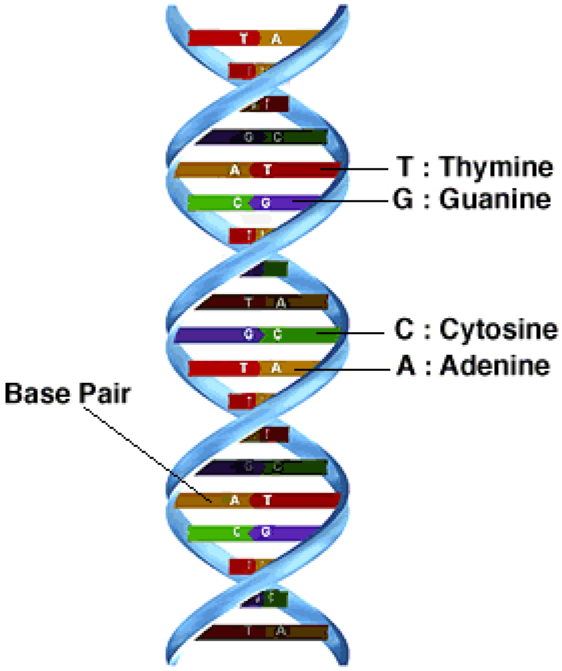


Figure 2.3 DNA, Helix Structure [22]

The sugar of DNA is deoxyribose from which it gets its name, while the sugar of RNA is ribose. The other difference between DNA and RNA is in their bases; the four bases of DNA are guanine, adenine, thymine and cytosine but instead of thymine, RNA has a base named uracil.

Each strand of the double stranded DNA has the ability to replicate itself and to encode information. The storage capacity of the DNA is extremely huge for an encoding device. 150 Mbytes of information can be encoded by one cubic micrometer of DNA.

A gene is a nucleic acid segment, which contains the information necessary to rule for any function in the organism. To be able to function, these ruling DNA segments enable synthesis of proteins, by coding for functional RNAs that has the sequence information of the proteins they encode for. The hereditary information encoded in the DNA, including the genes and the non-coding sequences, is called the genome of an organism. The human genome consists of approximately 3.2 billion of DNA base pairs [32]. These base pairs contain 20,000-25,000 genes. 1.5% of the whole human genome is only capable of synthesizing proteins; those protein coding regions are called exons. The rest of the genome is called junk DNA.

Proteins are the machines that control the function of all living things. Typical examples of their responsibilities are the catalytic activity, binding and transport. The function of a protein is dictated by its shape. The information needed to build up a protein is hidden in the cell genome which is composed of the genes. Each gene which is a unique sequence within the DNA, is basically an instruction manual for the directions to synthesize a protein. On the other hand it also has the responsibility to decide under which conditions, which proteins, through which cells should the synthesizing process be done, according to the information embedded in it. This process of transformation of the gene to the protein is called the *gene expression* and it occurs in two steps. The first step of the transformation is the phase which is said to be the *transcription.* In this step, the gene which is located in one strand of the double stranded DNA is used as a template for building up the RNA. This RNA is called messenger RNA (mRNA). In the second step which is called the *translation,* the RNA which had been constructed in the first step, is responsible for synthesizing proteins and the proteins are the ultimate products of the gene expressions. Proteins are known to be composed of amino acids covalently linked via peptide bonds. Three bases of RNAs are necessary to form a single amino acid. This gathering of reasonable three mRNAs is called a codon. Every codon over an mRNA corresponds to an anti-codon on a tranfer RNA (tRNA). There are twenty different amino acids and each can only be expressed by distinct combinations of anti-codons (or codons). This second step of the gene expression, occurs in the cytoplasm, within the ribosomes. The codons that are arrived to the ribosome, respectively call for charged tRNAs. A charged tRNA is the tRNA which is already connected to an aminoacid. These aminoacids bind to each other in a linear order. Those bound aminoacids are the products of gene expression (Figure 2.4 illustrates the two steps of the gene expression.). The linear sequence formed by binding of the amino acids is the primary structure of the protein. Through certain phases, a protein, later folds into its characteristic three-dimensional structure.

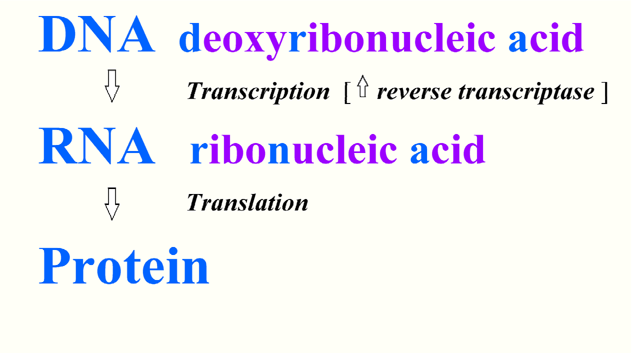

DNA deoxyribonucleic acid
Transcription [⇧ reverse transcriptase]
RNA ribonucleic acid
Translation
Protein

Figure 2.4 Gene Expression [18]

When the gene is transcribed upon instructions from certain signals within the cell, its mRNA will be present in the cell at that stage. mRNA within the cell show what genes are transcribed and the amount of transcripts at a certain stage of the cell. Microarray technology enables to determine these levels in a high throughput manner.

2.1.2 Microarray Experiments

For measuring gene expression, there exist various techniques. Certain examples for those techniques are differential display, northern blots and serial analysis of gene expression. In all of the techniques a complementary DNA (cDNA) strand is formed for both two strands of the DNA. This process is said to be hybridization which is a chemical reaction that forms a double stranded nucleic acid by joining two complementary strands of DNA and RNA. If it is a DNA-DNA match, the bases matching should be adenine-thymine and guanine-cytosine. (As mentioned in the previous sections RNA has uracil instead of thymine.)

For a microarray experiment two transcribed mRNAs from two distinct sources and a probe consisting of a chain of nucleotides are needed. There has to be enough probe sequence to give both sample 1 and sample 2 to bind with as much probe as they will and in both samples there may exists multiple copies of many genes. The aim is to detect which sample will have higher concentration of mRNA complementary. To be able to separate the samples at the end of the experiment, they must be pre-labeled. For the labeling fluorescent dyes are used; Cyanine 5

(Cy 5) and Cyanine 3 (Cy 3). Cy 5 is the green dye and Cy 3 is red. Both the sample dyed green and the sample dyed red should be mixed with the probe in parallel. The mixing will let the hybridization occur. To decide which sample has the higher performance to match with the probe sequence, the concentrations of the colors will be compared [2]. (The Figure 2.5 illustrates a microarray experiment)

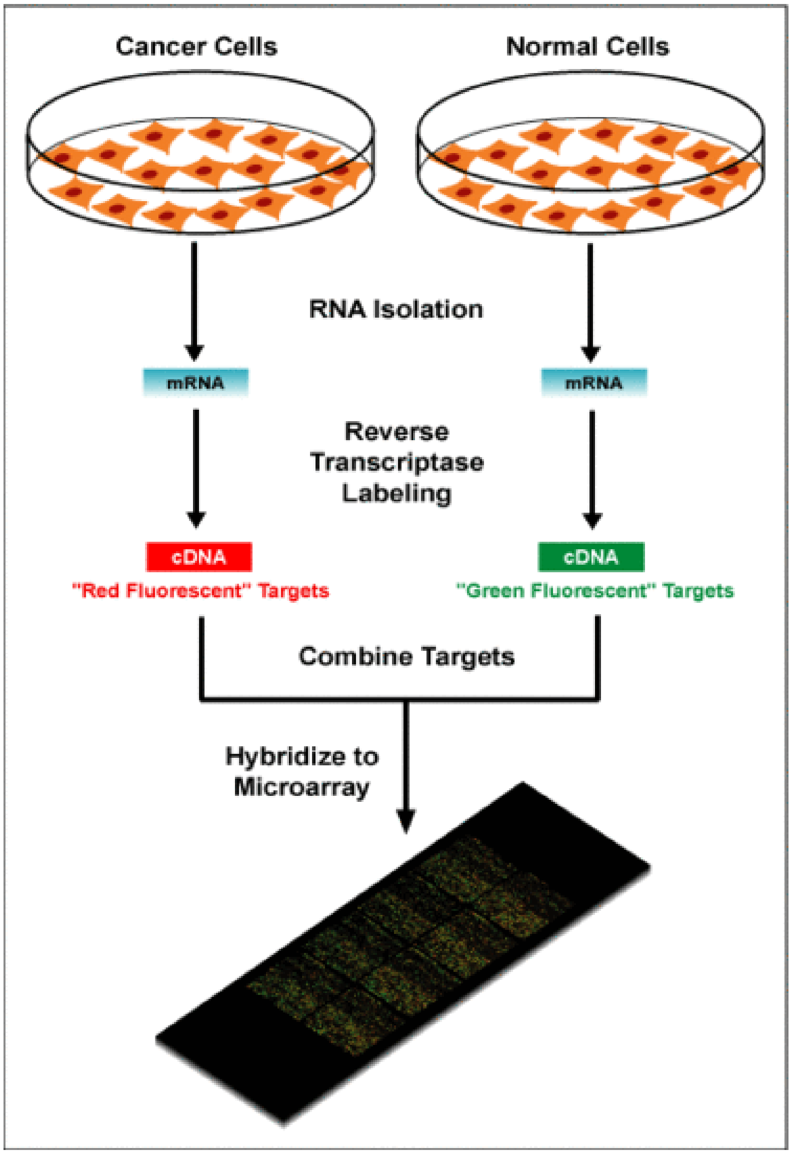


Figure 2.5 A Microarray Experiment [19]

2.1.3 Different Microarray Technologies

Various microarray technologies might be added up in two general titles.

- *Spotted microarrays* in which generally the probes are either oligonucleotides or complementary DNA (cDNA)s. The probe is spotted on to the surface of the microarray. Each spot to reflect different gene expression levels. The hybridization is done with cDNA labeled either green or red, to be able to identify the diseased tissue or the normal. As a result the differences between the colors with respect to each other are acquired.

- The *oligonucleotide microarrays* give the absolute value of gene expressions and use the probes to match the sequence of known mRNAs. To make comparison of different genes exhibiting character, two separate microarrays should be used. Affymetrix microarrays are the example of oligonucleotide microarrays.

2.1.4 The Usage of Microarrays

In *Differential Gene Expression Studies* the genes are examined under distinct experimental conditions like comparing the organisms in different development stages or genes in different tissue types. A very typical example of differential gene expression study is comparing the gene of normal tissues with the gene of the diseased [2].

A very similar study of differential Gene Expression is *Gene Co-regulation Studies* which compares the profiles of the genes with each other. The aim is to select the genes that show difference under variant conditions, coordinated or correlated with each other. So that the gene co-regulation studies, is an experiment which is made with two or more genes [2]. Correlated genes are considered to be on the same pathway. Through such studies one can determine which pathways are affected by the disease.

The *Gene Function Identification Studies* deal with the novel genes. The expression level of the novel genes is examined under different experimental conditions to be compared with a prescienced gene's expression levels under the same conditions. The known genes acting considerably similar to the novel gene are the clue for determining the function of the new gene [2].

Taking same gene from same source at different period of time and comparing the gene expression level differences is called *Time-Course Studies* [2]. Time course studies are similar to correlated gene studies in a way they both help to find which genes are correlated and which pathways are triggered by the disease.

Studying over a sample, a tissue or a patient's reaction to an exposure of different dosages of a chemical (mostly drugs) is called the *Dose-Response Studies* [2]. Effected genes reveal which pathways this drug shows its effect on.

Identification of Pathways is trying to disclose the route like which genes and which products of the genes function in which cells. *Identification of the Gene Regulatory Networks* is important because of the gene regulatory network's specialty of controlling the gene expression [2].

Predictive Toxicology Studies are based on microarray databases storing huge amounts of involved organs and their response to specific toxic agents. In such a study, the goal for the pharmaceutical industries, is to identify toxic influence of an unapprehended compound as seasonable as possible [2].

As they have the ability to uncover the expression patterns which have distinguishing traits for particular diseases, the gene expression experiments of microarrays are important for *Clinical Diagnosis* [2].

In *Sequence-variation Studies* the goal is to reveal the sequence variations of DNA which is correlated with the phenotypic changes [2].

Our experiments in which real world data is chosen, three different datasets presenting the expression levels of genes obtained from cancer and control patients is used. Initially a colon cancer data which consists of 62 tissue samples comprising of 40 tumor, 22 normal, each with 2000 distinct gene expression levels. This data set is obtained from the usage of Affymetrics oligonucleotide microarrays by Alon *et. al.* [15]. Secondly an ovarian cancer data set obtained from Petricoin *et. al.* is used. This data set contains 162 tumor and 91 normal tissue samples, each represented with 15154 gene expression levels [20]. Finally a prostate data set covering 12600 gene expression levels for 52 cancer and 50 control genes is used. This prostate data set is obtained from Singh *et. al.* [21]. Microarray experiments were used by the researchers, to obtain the three data sets of expression levels.

As a result of microarray experiments, very large amounts of expression levels of genes are achieved from the simultaneous works from thousands of genes. Therefore, although it is extremely important to use these datasets for accurate classifications for diagnosis and treatment of relevant diseases, it is not very easy to succeed [1]. Recently scientists are concentrating on this subject. There are a number of proposed solutions for this problem [3, 4].

2.2 Feature Selection

It is important to represent the data in a range of features in pattern recognition problems. In terms of time and cost, it is not profitable to gather and handle all the data available [2, 5].

Therefore in machine learning, *Feature Selection* methods are frequently used. Feature Selection which is also said to be *subset selection*, is basically eliminating the irrelevant and redundant attributes and selecting a reasonable subset out of the entire data. This selection of the plausible features among the whole extensive data may have an impact which is considerable over the effectiveness of the resulting algorithm [5, 6, 7].

There exist two different approaches for the feature selection problem in the literature. These approaches are called filter approach and wrapper approach.

In *wrapper approach,* the classifier system is trained with the feature subset as an input. In this method, the variable subsets are selected by the classifier are assessed by the learning algorithm. The problem with this method is known to be the large amount of computation that it requires because every time the classifier should be retrained [8]. There are two ways for the wrapper approach. One of them is the forward selection and the other is the backward elimination. In backward elimination, the initial set consisting of all variables is diminished by the elimination of the less significant variables. Oppositely, larger subsets are formed by combining the variables in forward selection [37].

On the other hand, in *filter approach* features are selected before the actual learning algorithm that uses a predefined measure. Because of preselecting the features, selection method does not affect the performance of the learning algorithm [37]. In the literature there exist various approaches which computer scientists proposed about the feature selection problem and since the filter approach uses less amount of CPU, it can appreciably be seen in the most recent works [9].

The preliminary focus of the researchers for the feature selection problem was on the breadth first and branch and bound algorithms. The results were satisfactory with the conventional statistical classifiers, which are simple probabilistic classifiers but experiments with the non-linear classifiers gave destitute results [10,11]. On the other hand to perform effective feature selection heuristic search and randomized population based algorithms are also being tried. An example to these kinds of algorithms are Genetic Algorithms which made the biggest impact among other concerned examples [12,13].

In the literature, the problems that require searches within the microarray datasets are one of the most conspicuous examples that the Genetic Algorithms are used for. The microarray datasets compose very large search spaces because they contain very large number of feature and in addition there exists two opportunities for every feature in feature selection problem (the feature is selected, or not selected). Thus, in many approaches in which the aim is to filter out the significant features within the microarray datasets, genetic algorithms in combination with distinct classification methods are preferred recently.

2.3 Genetic Algorithms

The term Genetic Algorithm is abbreviated as GA. The principal that the GAs are based on was first proposed in 1960's. Those days the idea of *evolutionary strategy* was being transpired in Germany by Ingo Rechenberg and by Hans-Paul Schwefel and in USA. A very similar idea was being worked on by Lawrence J. Fogel naming his research *evolutionary programming*. In both of the approaches, the search is done using mutations and selections. The missing idea with these reseaches was considered by Fraser and Bremermann. That idea was the recombination. Those researches, including mutations, recombination and selection, enlighten the approach of John Holland's which is said to be the GAs [12,14].

The basic idea of the evolutionary programming and the GA is simply based on imitating the laws of the nature. The evolution is the model to be imitated. One of the basic ideas underlying the evolution is the natural selection. Charles Darwin is the natural scientist who proposed the idea of natural selection. According to natural selection, an increase in fitness of the organism and an increase in the organism's ability to reproduce are provided by the small heritable variations in organisms. The theory of natural selection is set out in detail in Charles Darwin's book called *The Origin of the Species* which is published in 1859 [23]. After inheritance based on genetics is discovered by Gregor Mendel, in 1930's the combination of these two approaches gave the evolution its modern shape. Evolution is known to be the variations from one generation to next during recombination. The nature follows the Darwinian principle of *survival of the fittest* and mostly selects the fitter ones rather than less fits to let them to reproduce [25]. The variations occur among the gene expressions of the parents and the offsprings. This variation is achieved mostly from the process of *cross-over*. During cross-over the chromosomes (a single large DNA molecule containing many genes) from each parent are paired to exchange some pieces of their genes as shown in the Figure 2.6. In addition to this, the *mutations* which are the random changes in the base-pair sequence of the genes, may also rarely lead to genetic heritable variations.

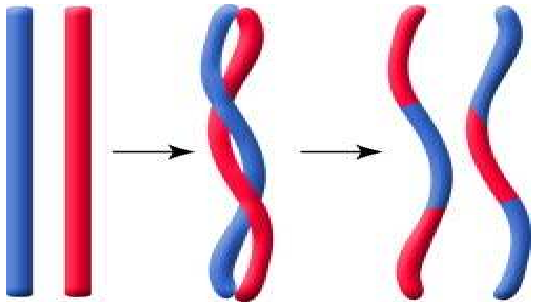


Figure 2.6 Crossing-over Chromosomes [24]

The processes of cross-over and the mutation which were mentioned in the previous paragraph, are called the *genetic operators* within Genetic Algorithms (GAs). In the basic GA, initial population is constituted from a set of individuals; this population is transacted with the genetic operations to form offsprings. This procedure is continued from one generation to another by replacing the individuals of the concerned generation with the offsprings. So that, the offsprings belonging to the related generation, will be the parents of the next generation. The only need of a basic GA is a function to calculate how suitable the offsprings are [14]. This function will generate the solution for the use of GA to behave relevant to the Darwinian mentality of *survival of the fittest* and imitate the environment to decide how fit the solutions are. So GA will be able to use the historical information achieved from the previous generations and speculate on the offsprings which are expected to have better performance with respect to the previous generations [26]. This function is called the fitness function. A scheme for the basic GA is shown below (Figure 2.7).

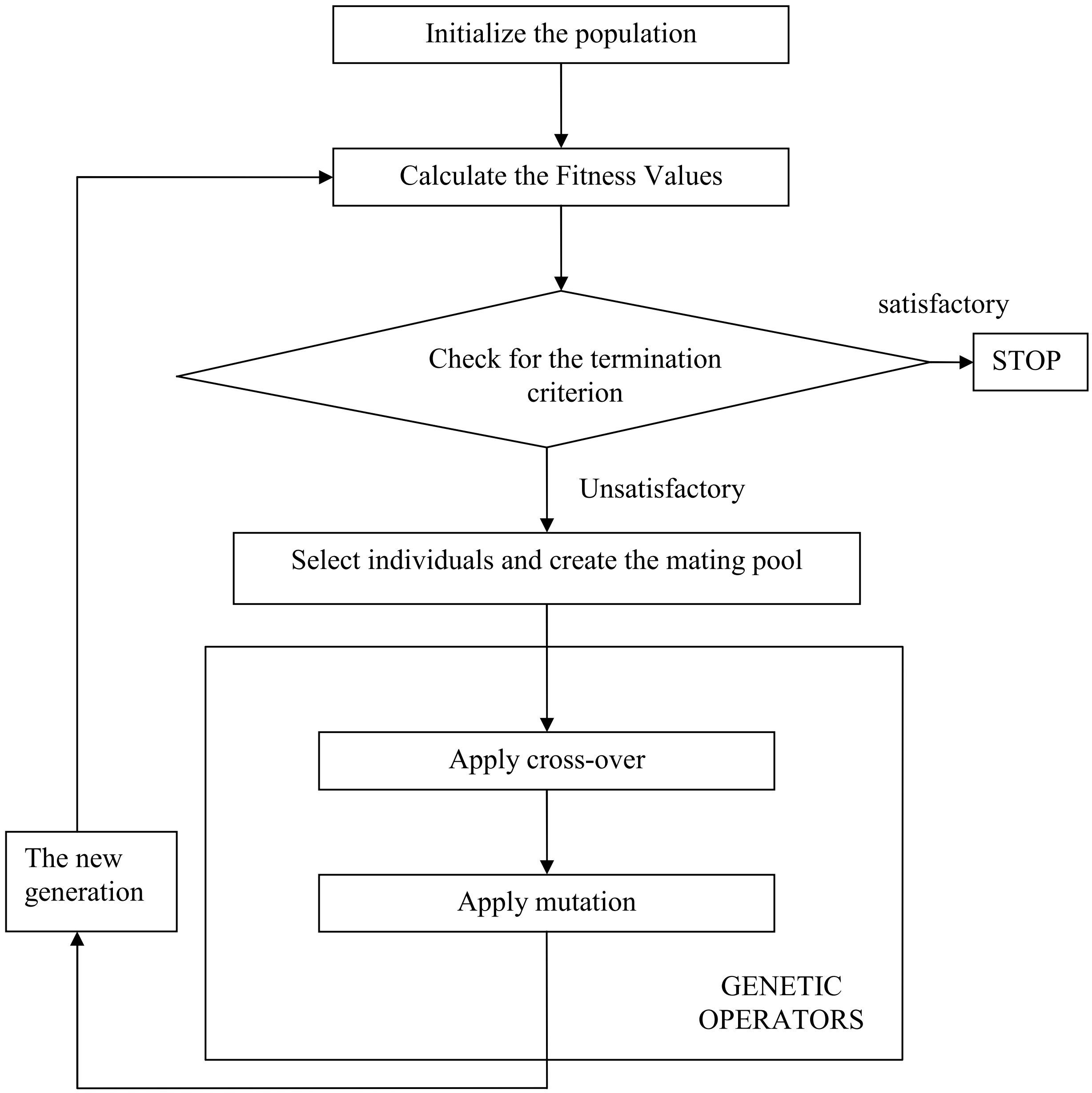

Initialize the population
Calculate the Fitness Values
Check for the termination criterion
satisfactory
STOP
Unsatisfactory
Select individuals and create the mating pool
Apply cross-over
Apply mutation
GENETIC OPERATORS
The new generation

Figure 2.7 A Basic GA

2.3.1 Encoding

In a GA, there are several coding schemes to represent the genes in the individuals. The preference about the encoding technique is made according to the related problem. One of the most used techniques is *binary encoding*. This technique is popular because the first works for GA is done using binary encoding and certain patterns can be followed. In binary encoding, an individual will be formed of a string that consists of 0s and 1s, and it will be represented as {10011,00101,….. ,11001}. Another way to encode the individuals is *permutation encoding* which is generally used in ordering problems. The idea in such encoding is based on trying to find the minimum path. The individuals have a string consisting of numbers and one individual will be represented as {1,7,5,….,9}. In permutation encoding the search is generally done using mutations, cross-over is generally not used. *Value encoding* is another technique, in which the real values are used to express thee individual. A formula that signifies the problem is arranged. The solution of the equation coming from a candidate parent is hoped to be zero. The surviving probability of the individual is calculated according to the result taken from that individual with respect to the equation. So an individual with the solution that is closest to zero will have the highest probability to survive. Another way to encode the individuals is called *tree encoding*. In this technique, a tree is used to represent the calculations. Error for every point is calculated and the point with low error has higher fitness value. In this technique, the cross-over is done among the branches and each node can be changed during the mutation. An example of this type of encoding is shown below (Figure 2.8) :

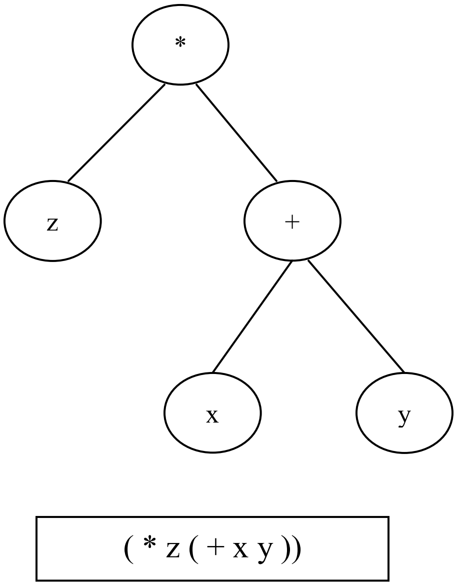


Figure 2.8 Tree Encoding

2.3.2 Initialization of the Population

A population is formed by a set of individuals. The initial population is usually generated randomly. Usually the population size is kept constant but depending on the aim of the algorithm, in the future generations the number of individuals in a population may be allowed to vary. The fitness evaluation function is again chosen with respect to the problem. The individuals that are highly fit according to the fitness criterion will have more probability for getting selected as a parent to reproduce the children. The parents are chosen and form another population under the name of mating pool [25], so that the parent candidate with a high fitness value will be able to have more copies in the mating pool. By this means, the knowledge from the high fit individuals will be more effective over the next generations. For the constitution of the mating pool mostly Roulette Wheel Parent Selection or Linear Selection is used [25]. To the parents selected with respect to their fitness values, genetic operators are applied.

2.3.3 Fitness Function

This function decides how fit the individuals are. The character of each individual is evaluated at this part. So that, this function is be called as the objective function of GA. Each individual is taken as an input, and according to the individual's ability to solve the problem, the fitness values are calculated. At the end, each individual is calculated to have a single numerical value to represent its fitness value. The fitness values returned by the objective function are used to calculate survival probabilities of the individuals depending on the selection scheme. Individuals having higher fitness values will have the right to survive for the next generation, others are eliminated. This idea is directly based on the nature's survival of the fittest principle.

2.3.4 Cross-over

As mentioned before, while cross-over is applied to two randomly selected parents, the main goal is to exchange the information between selected individuals. As a result the offspring(s) is generated with the combination of the knowledge acquired from these two parents. There exist different cross-over techniques. Some of them are *one-point cross-over*, *two point cross-over*, *multiple-point cross-over, cut and splice* and *uniform cross-over* [25]. In one-point cross-over, a point is detected within each parent and starting from those points the strings (so that the information) is exchanged (Figure 2.9). In two-point cross-over, instead of one point for the exchange, two point are determined and the region in between the points are swapped (Figure 2.10) while in the multi-point cross-over, as following the same principle, more then two points are chosen. On the other hand, in cut and splice technique, two distinct points are chosen for each parent and because of that the children possibly end up with different number of genes as illustrated in the Figure 2.11. In uniform cross-over, as it is in the other approaches two parents may form two offsprings. In this technique, every gene (or every bit in each gene depending on the structure chosen for the population of GA) of two parents are compared in one by one with each other. The genes that constitute the offsprings are selected from each parent according to a uniform distribution.

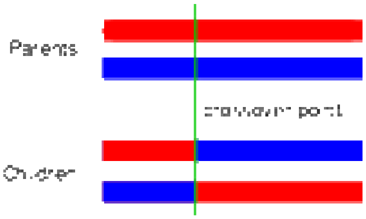


Figure 2.9 One-Point Cross-over [27]

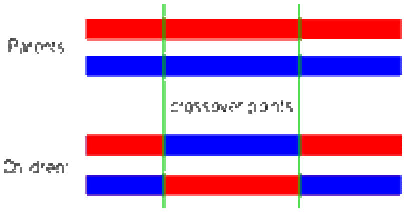


Figure 2.10 Two-Point Cross-over [28]

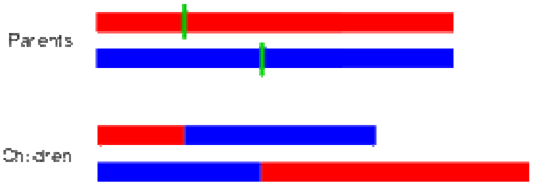


Figure 2.11 Cut and Splice Cross-over [29]

Depending on the chosen cross-over technique, an optimization for the parameter of the cross-over rate might be needed. This parameter stands for deciding what percentage of the two parents (that are matched for the recombination to occur and generate children) will be exchanged. Too high cross-over rate might cause loss of information gained and generally mutation rate is chosen around 10% in the literature.

2.3.5 Mutation

Final operation of a basic GA is mutation. The goal of this genetic operator is to provide random genetic diversity. Algorithmically it lets the code to search a distant area apart from the region that it is concentrated on. The information lost in the previous generations might be regained with the help of this random search. The idea is basically, selecting a random individual among the new generated children and changes a randomly selected gene in that child. The mutation rate is generally kept low with respect to the cross-over rate. A relatively high mutation rate would cause loss of information and change the GA in to a random algorithm. In general there are three mutation techniques. One of them is called *one-point mutation.* When encoding technique is binary encoding, one bit is chosen to be changed from 0 to 1 or vice versa. This selected to be replaced is called the mutation point. In *bit mutation*, which is another mutation technique, every bit in the individual which is decided to be mutated is changed from 0 to1 or 1 to 0. The last mutation technique is called the *uniform mutation.* In this technique a template individual is generated for the individual who is decided to be mutated. This template has the same length with the individuals. This template reveals which bits (genes) of the individual will be changed. An example for uniform mutation is illustrated in Figure 2.12 :

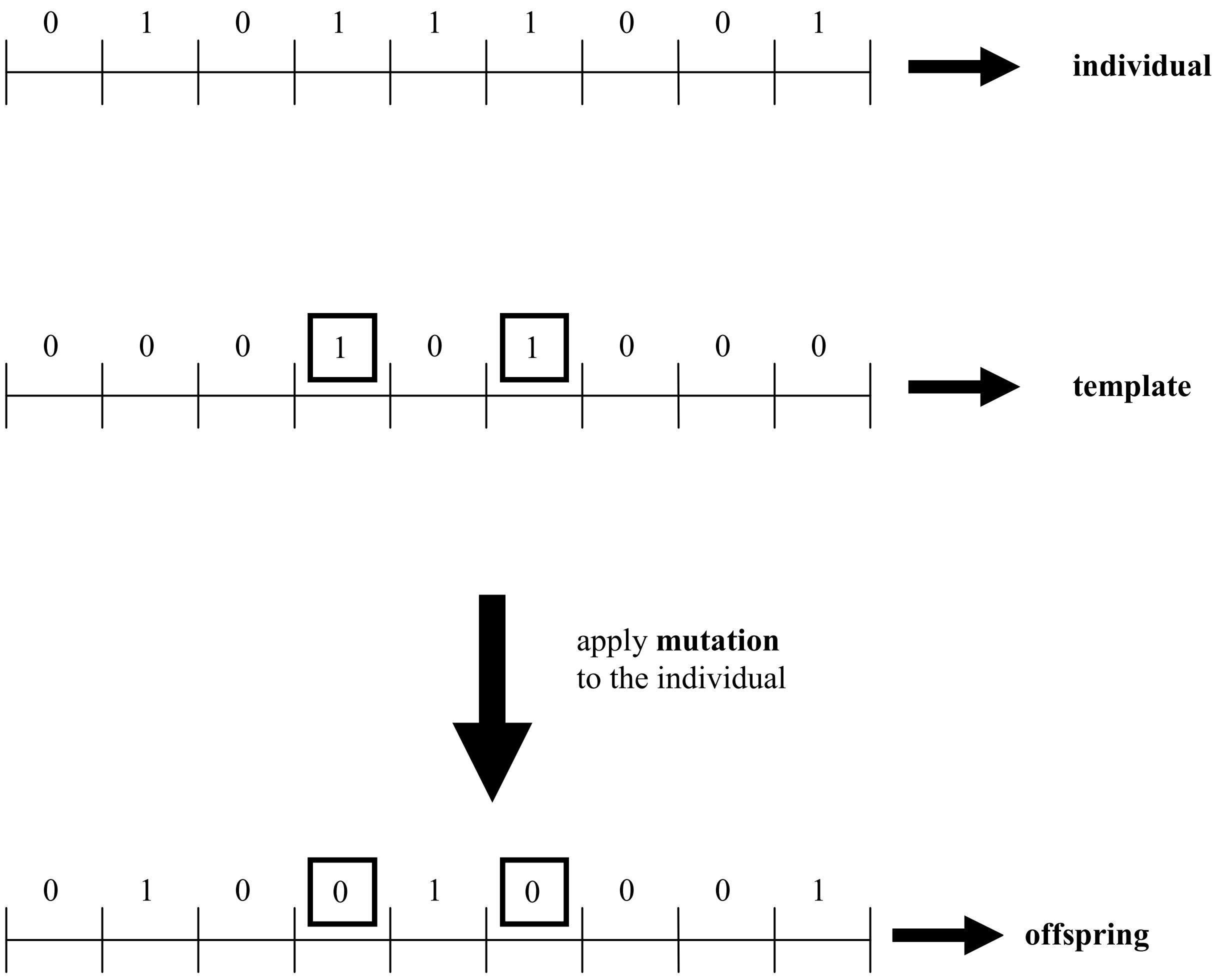


Figure 2.12 Uniform Mutation

3 METHODOLOGY

In our approach, we used genetic algorithms with roulette wheel selection for the feature selection problem. We try to select minimum number of features that are relevant for the classification of gene expression data among three different datasets obtained from oligonucteotide affymetrix microarrays. The datasets are mentioned in the previous chapter.

3.1 About the Algorithm

3.1.1 Data Structure

Generally in the literature binary encoding is preferred to be used for the data structure. But we preferred to store the indices of the features (genes) that exist in an individual, in to the array of that individual. The reason for our preference is the efficiency of the search within an individual. There exist two other different variables for an individual. One of them, which is named *FeatureCount* is used to store the total number of genes that are owned by the individual and the other stores the fitness value of the individual within the generation, so called *Fitness*. Zero-based indices are used to name the genes. This means the first gene is indexed as 0, the second gene is indexed as 1, and the third one is indexed as 2 and so on. According to above information, the parameters, for an individual in one generation, are exampled below:

```
double m_dFitness=1.0;
int    m_nFeature=3;
int    m_nFeatures[nMAX_FEATURES];
                      |
                      ↓
       m_nFeatures[0]=7;
       m_nFeatures[1]=39;
       m_nFeatures[2]=125;
```

The indices for the genes, belonging to an individual, are placed in a sorted formation. Every individual may have different number of genes and the algorithm is capable of crossing over the individuals having different lengths.

3.1.2 Random Number Generator

In our program, a very recent version of the *Mersenne Twister (MT)*is used to generate random numbers. MT which is a pseudorandom number generator, is developed in 1997, by Makoto Matsumoto and Takuji Nishimura [31]. Its running time is much faster then the other statistical generators and because of this reason it is mostly used in the statistical problems. The algorithm has the period of $2^{19937} -1$ and has 623-dimensional equidistribution property. These properties make MT the best random number generator which is implemented. The version that is preferred in our approach has just been released in March, 2005 and it returns the random number directly as a double number. This expedites the run time of the entire program by not loosing any time with the conversion of integers to doubles.

3.1.3 Calculation of the Fitness

As mentioned in the previous chapter, our GA requires an external algorithm to calculate the fitness values of the individual. In our work, we decided to use Support Vector Machines for this purpose.

Support Vector Machines (SVMs) are commonly used for classification and regression. The basic aim is to classify the items that are similar in their feature values. These supervised learning algorithms are known to be enhanced from linear classifiers. The input vectors are mapped to a higher dimensional space and data is separated with a hyperplane. This hyperplane puts the data in to two distinct classes. To both sides of this hyperplane which shows the border of different classes of the data, two hyperplanes in parallel are also invented. The generalization is known to be better as the margin between two parallel hyperplanes is larger. Thus, the distance between these two parallel hyperplanes, is aimed to be maximized, while the effects of the classification error is minimized [33]. The algorithm for SVM was originally proposed by Vladimir Vapnik in 1963 (see Figure 3.1)

.

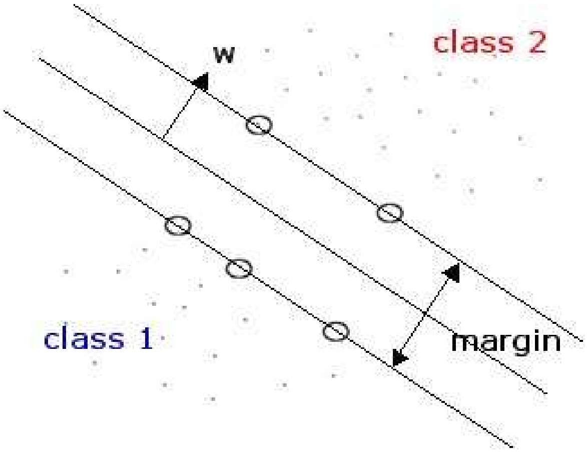


Figure 3.1 The margin between two classes achieved from SVM [36]

3.2 The Flow of the GA

Several trials with various combinations of the parameters are done in order to be able to decide which combination to be used. All of the data is considered for this optimization of the parameters. The constant parameters are decided to be as follows. The maximum number of genes (features) that an individual can have within a generation is limited to 30 (This parameter will be called as *Maximum Feature* in the following sections.). The mutation rate was decided to be 0.1.

A possible problem with the previous work of our project might be the inefficient exploration of the search which may be responsible for the loss of the genes that were important for the classification, in the very beginning of the algorithm. For that reason, these genes would possibly not have the chance to represent their classification powers in the further generations. Our approach overcomes this problem by assigning a parameter set for the minimum occurrence of every gene in the search procedure that controls each feature to be assigned to more than one individual. To be able to perform this, in a loop, for each gene we assigned two random numbers which are representing the individuals they are going to be set to. A suitable individual for a particular gene is decided on the basis of two constraints. The individual that is randomly selected for the related gene to be assigned, should own less than 30 genes. The second control is done by checking through the selected individual, to see whether it had already taken the related gene in the consideration or not. If both conditions are satisfied, the gene will be assigned to that randomly selected individual. Otherwise, the algorithm will look for the succeeding individuals one by one, until it finds a suitable individual to place the gene. Thus, a homogenous distribution of genes over the individuals is expected. The constant number chosen for this parameter affects another variable which is the maximum size of the population. This is important for the decision of the optimum number for the population. After a number of tests, the constant number is chosen as 2, and this parameter named *Feature Repeat*. This number makes sure that every gene is represented at least in two different individuals and unlike the previous approach; it gives a survival chance to every gene.

The population size is the variable that stands for the number of individuals that are going to exist in the initial population. Before the GA is initialized this number is calculated according to three facts. The population size is directly proportional to the total number of genes in the dataset and the number of repeats we want each gene to appear in our population. On the other hand, it is inversely proportional to the parameter standing for the maximum number of genes that each individual might have. This parameter is called as *Maximum Feature* within the formula. The formula used in our algorithm, for the population size is as follows:

$$\text{Max Population} = \frac{\text{(Feature Repeat)} * \text{(Total Number of Genes in Dataset)}}{\text{Maximum Feature}}$$

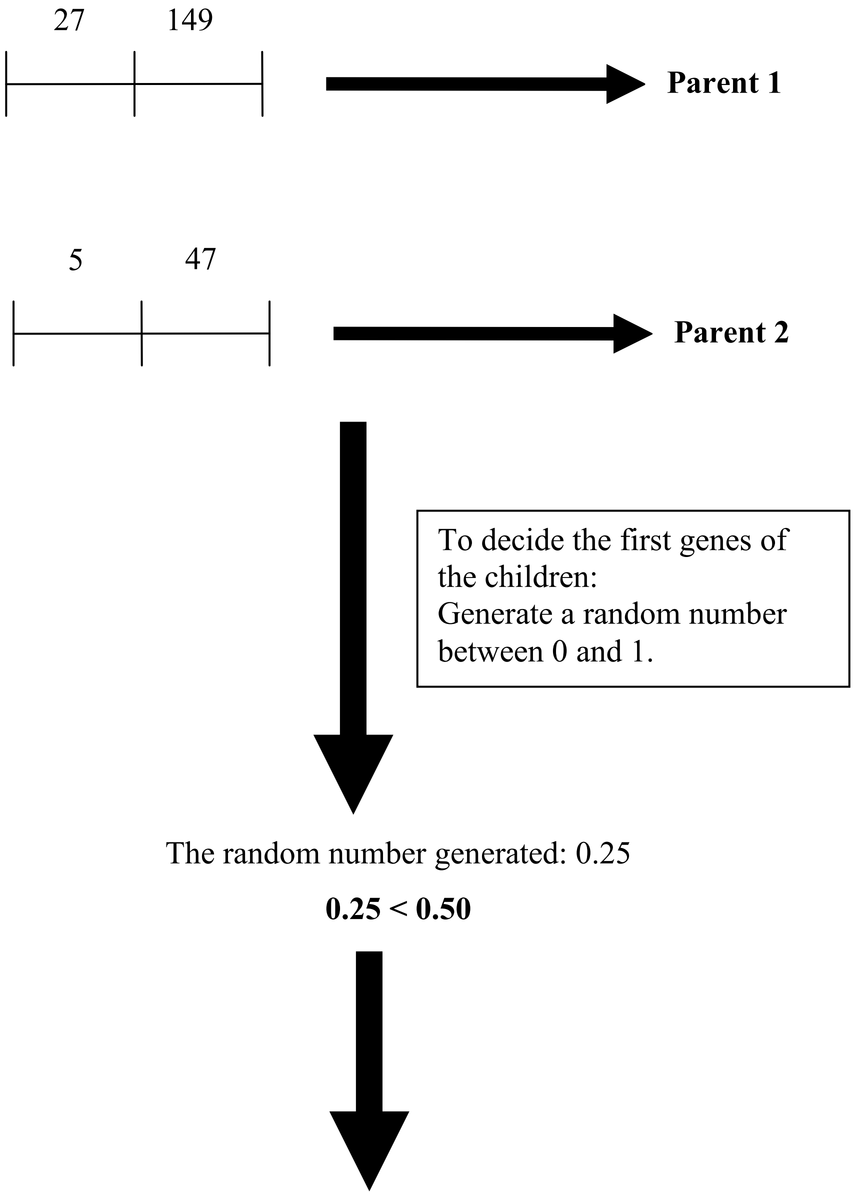
(The result is rounded to the integer above not to loose any data from the Maximum Feature.)

The initial population is generated randomly according to the parameters above. The algorithm consists of two distinct parts. The first part is made up of runs contributed to *search the fitness space* to generate fitness values of all the genes and the second part is the *main algorithm* where the feature reduction is achieved while maximizing the classification accuracy

3.2.1 Runs for Searching the Fitness Space

To be able to search the fitness space a certain number of initial runs are done. The goal of this part is not to find minimum set of features but to establish a foundation for the second part at which results will be obtained. In this part the algorithm aims to search the space and to assign fitness values to each gene separately. For this purpose, a population consisting of all the genes, which is already generated, will be used. The goal is not to protect any gene or individual with regard to its fitness value, but to collect the fitness values of each gene in combination with maximum number of different, random genes. For this kind of selection, a standard random selection is applied. Mersenne twister is decided to be used as the random number generator. As a result each gene will have the equal opportunity to exhibit its character in combination with various genes. The fitness values are calculated to be stored. Depending on this idea, a cross-over technique which does not destruct any gene, is selected and no mutations are done in this step since it is not necessary.

As the cross-over technique, uniform cross-over is used in this part of the algorithm, randomly two parents are selected among the individuals. Since mating is not decided up on the fitness scores, each parent is mated once in one generation. The genes are stored and ranked in an individual. Each gene for the children is selected with either from the first parent or the second according to the probability of 50%. For this process, for each gene a random number is generated between 0 and 1, if the result is less then 0.5, first parent’s related gene is assigned to the first child and the second parent’s related gene is assigned to the second child’s related gene. Otherwise exactly the opposite assignments are done. The cross-over in these initial runs is expected to give the maximum variations related with our aim, to enable the usage of all genes with different combinations. For this reason, the cross-over rate is selected as 50%. Below, the cross-over technique used in this part of the algorithm is exampled using 2 parents consisting of 2 distinct genes (Figure 3.2).



To decide the first genes of the children:
Generate a random number between 0 and 1.

The random number generated: 0.25

0.25 < 0.50

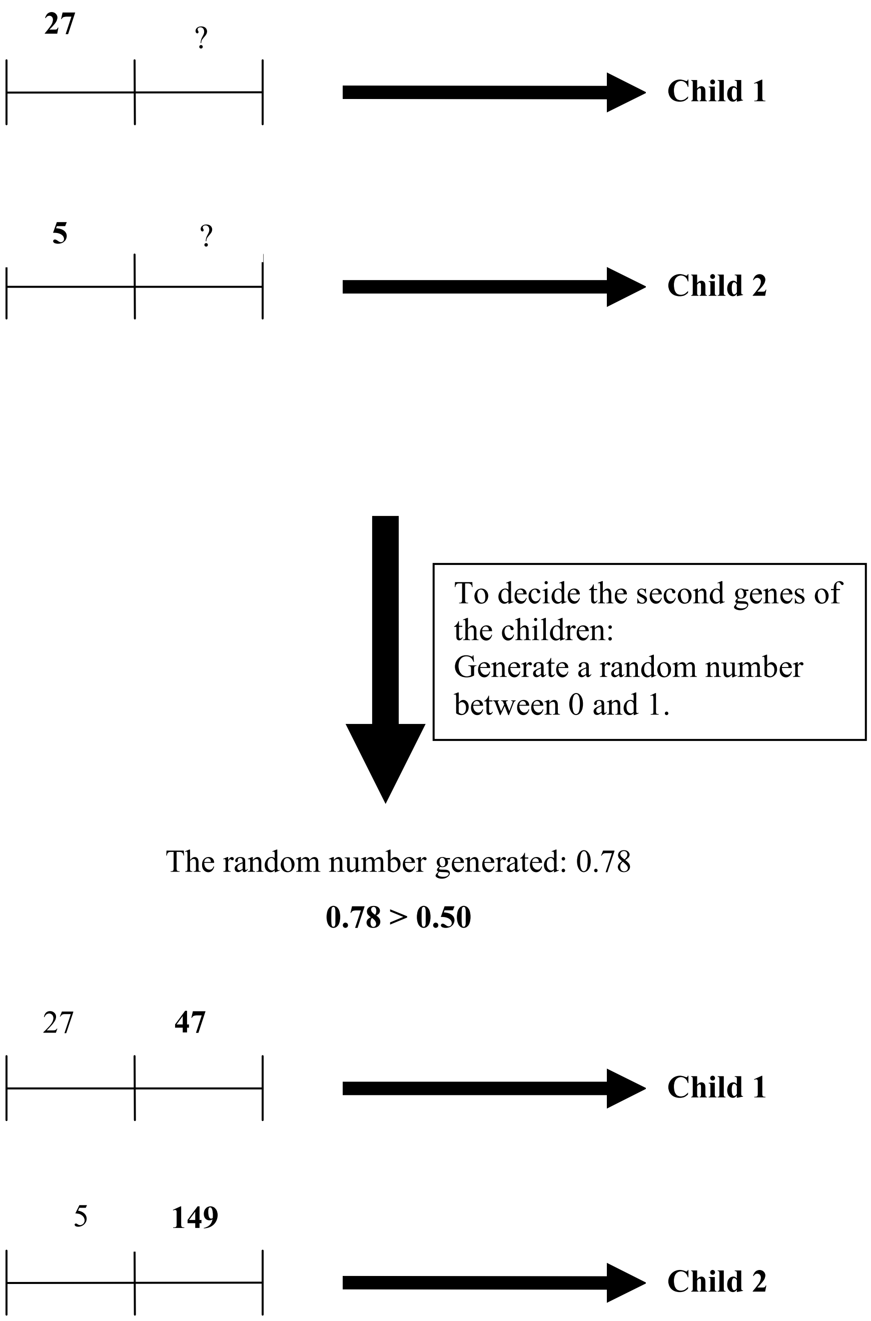

27
?
Child 1
5
?
Child 2
To decide the second genes of the children:
Generate a random number between 0 and 1.
The random number generated: 0.78
0.78 > 0.50
27
47
Child 1
5
149
Child 2

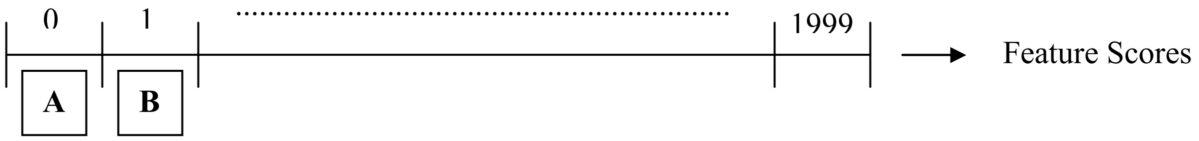
Figure 3.2 Cross-over in the Runs for Searching the Fitness Space

Another parameter to be chosen is the number generations to be run for the initial phase of the algorithm. We tried several values for this parameter such as 10, 20, 50 and 70. As a result, it is observed that the more the number of the generations increased, the earlier the solutions are achieved in the main algorithm. The gap between the trials 10, 20, 50 was more conspicuous, so initial search step is decided to be run for 50 generation.

3.2.2 The Main Algorithm

At the end of the initial run we acquired considerable knowledge about the characters of the genes. In each generation, the average fitness scores of the genes are taken and added to average fitness score of that gene from previous generations for their use in the next step; the main algorithm. The fitness score of a gene is the sum of the average fitness score of that gene for every generation.

As the data structure of the main simulation the fitness coefficient of the genes and the array for the fitness values of the individuals are used. The fitness coefficient of the genes stands for average of the summation of the fitness values of the individuals that contain the related gene [30]. Below, the array is shown schematically for colon dataset having 2000 genes (Figure 3.3) :



first individual indexed “0” → { 0,1,437,1587} → Fitness Value = x

second individual indexed “1” → {1,765,1154} → Fitness Value = y

A (Gene 0) = x

B (Gene 1) = (x+y) / 2

Figure 3.3 Calculating the Fitness Values of the Genes

This value is calculated as the average fitness score for each gene in each generation and added to average fitness of that gene from previous generations. Thus, as the generations progress, the genes that are highly used, will have the chance to make their fitness values larger. Those genes which have larger fitness values definitely have higher probabilities to be selected by the roulette wheel selection algorithm. The fitness values of the individuals that involve a certain gene are averaged within the generation so that is, the dramatic gap that might occur between the fitness values of the genes if they are summed could be eliminated.

Although in the literature it is possible to see studies destroying all the population in certain number of generations, converging to significant results [30], in our test runs we saw that not destroying the population yields to better results. Thus, we decided not to destroy the population. Destroying the population means destroying the accumulated know-how; the only remaining thing will be the feature scores. But it seems to be better to continue from the existing point and try to improve it until convergence.

To select the crossing-over scheme various methods with different cross-over rates are tried. We ran the algorithm for 3000 generations to be able to see the differences between the cross-over methods clearly.

The graphics acquired from the trials of three different cross-over methods is shown below, in Figure 3.4:

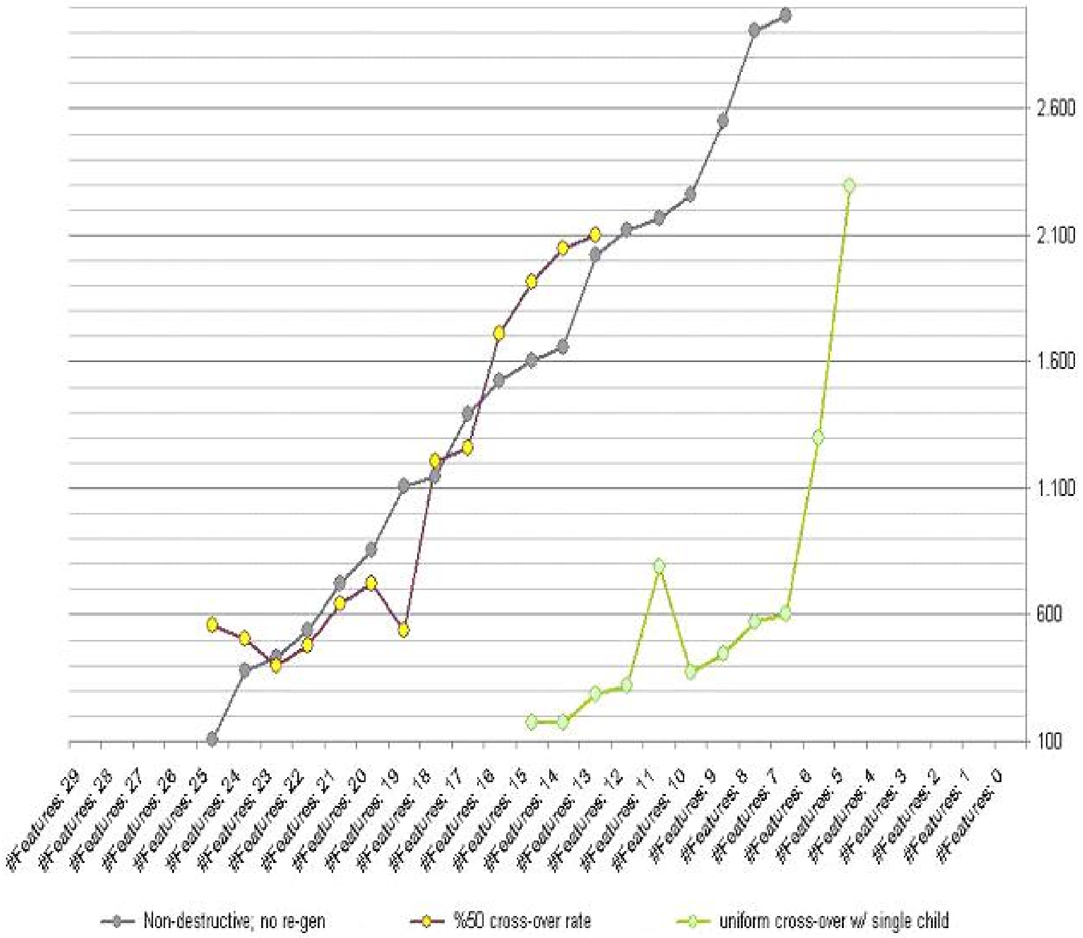


Figure 3.4 Comparisons of Some Cross-Over Techniques

In the above graphics, the vertical axis stands for the generations that the algorithm ran for and the horizontal axis stands for the number of features that the algorithm could filter out. The dots represent the solutions achieved in the corresponding generation using the corresponding number of feature with the accuracy of 100%.

The first trial which is symbolized with the color grey in the chart, was using the cross-over rate 20% and the second trial which is colored yellow in the chart, was using the cross-over rate 50%. As it can be seen above, we could not detect any obvious differences. Then we tried uniform cross-over generating single child and this technique significantly improved the performance of the algorithm in terms of time and precision.

Therefore, we decided to use uniform cross-over in which two parents matched to constitute a single child. When a single child is attained from two parents, the variation in the search is increased and as we are searching within huge datasets, this kind of cross-over technique is suitable for our work. In this method to be able to generate a single child, two parents are selected among the population while the ones with higher fitness values are favored. So that, the relevant parents would have more chance to evince themselves and the size of the population is avoided from being reduced through every generation.

In the uniform cross-over ascertaining a single child, two parents are chosen with the roulette wheel selection and matched. Every single gene of the child is chosen either from the first parent or the second with the probability of 0.5. If the sizes of the two parents are not equal, the parent having the least number of genes is maximized by using one or more gene of that parent more then once.

The genes within the significant parents are highly used and so have higher fitness values. This means as the generations go further, the parents having same genes will have higher probability to mate. The genes are stored sorted within the parents and so same genes might be in the same place within two different parents. So that, either on or the other gene will be selected for the child and the child will have no identical genes. But on the other hand, when same genes are placed in different places of two parents, the child may contain two identical genes at the end of cross-over. In that situation, the identical genes are merged and so the size of the child will be reduced.

Below, each possibility is illustrated. Figure 3.4 demonstrates an example of match of parents having identical genes in the same order, while Figure 3.5 demonstrates having identical parents in different order.

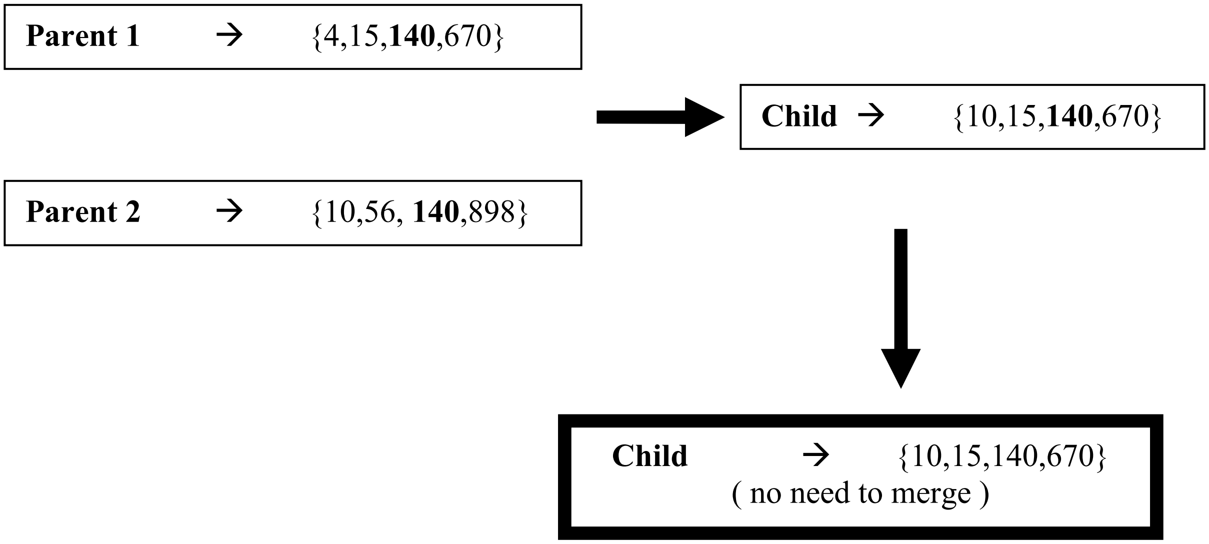

Parent 1 → {4,15,**140**,670}
Parent 2 → {10,56, **140**,898}
Child → {10,15,**140**,670}
Child → {10,15,140,670}
(no need to merge)

Figure 3.4 A Match Causing No Shrink

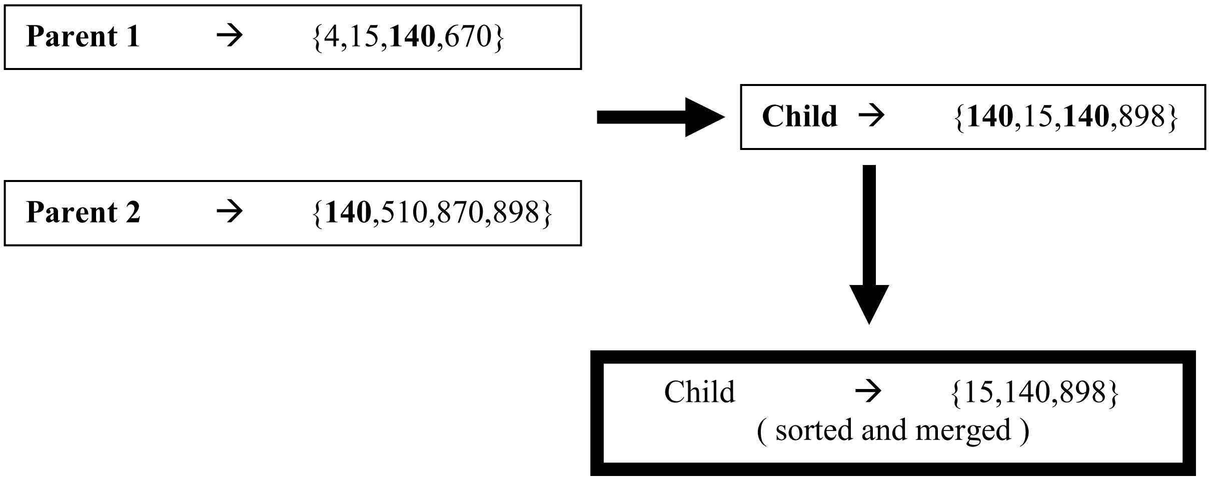

Parent 1 → {4,15,**140**,670}
Parent 2 → {**140**,510,870,898}
Child → {**140**,15,**140**,898}
Child → {15,140,898}
(sorted and merged)

Figure 3.5 A Match Causing a Shrink

The next genetic operator applied to the algorithm is mutation. In mutations a chaotic element is injected in to the algorithm in low ratios. This provides the possibility to search a random point or points apart from the space that the algorithm is concentrated on. Deciding to mutation rate is a delicate thing to do, for the reason that high mutation rate can cause big jumps from the existing solution which leads to information loss and eventually leads to divergence. If the dataset used is large enough, then the rate could be relatively large. In this work mutation rate is tried to be 5%, 7%, 10% and 20%. The best solution is taken from the tests done with using the mutation rate 10%. For the mutation step, for each gene of an individual, a random number between 0 and 1 is generated. If the number is larger then 0.1, then the related gene is obliterated and instead randomly a popular gene is chosen by roulette wheel selection.

After the mutation a new generation is created and the fitness values of the new individuals are calculated by SVM. A constant number of worst individuals of new generations are replaced with same number of best individuals of the previous generation. Those individuals with the best fitness values of the old generation are called the *elitist parent.* We tried the elitist parent ratio as 10% and 20%, 10% gave the optimum results. Higher ratios for the elitist parent is to become conservative that, the more you do not risk to loose the information gained, the more you can not find betters. In further generations the number of individuals which have the fitness value 100% will increase. In our algorithm, the ones having the least number of genes among those highest fitness valued individuals have the first priority to be chosen as the elitist parent. Accordingly, in the children, the ones having the highest number of genes are replaced. This helped the algorithm to work faster and so converge to better results.

4 RESULTS

4.1 Datasets

In our approach, we have tested our genetic algorithm implementation with real world data as mentioned in the previous chapters.

- The first data used is a colon tumor dataset obtained by Alon *et. al.* using Affymetrics oligonucleotide arrays, exhibiting gene expression levels of 2000 genes for 40 tumor and 22 normal colon tissue samples [15].

- In the second dataset which is obtained from Singh *et. al.* [21], consists of 12600 gene expression levels of 52 prostate cancer data and 50 control patient data.

- Third dataset contains ovarian cancer data with the gene expression levels of 15154 genes taken from 162 diseased and 91 control patients. This data is obtained from Patricoin *et. al.* [20].

4.1.1 Previous Approaches Using The Same Datasets

These three datasets are worked on, by various researches to be able to achieve the minimum subset of features (genes) that classifies the entire data. Although there were some other approaches to this problem using the same datasets like Liu *et. al.* [34] and Bing *et. al.* [35], the best results in the literature were gained by Küçükural *et. al*. [9]. This work was the previous approach of our group. Küçükural *et. al.* was able to classify the colon cancer data with 98.38% accuracy using 12 features, ovarian cancer data 100% using 12 features and prostate cancer data with 96.7% using 19 features.

The best results up to 2002, for ovarian cancer dataset was achieved by Lui *et. al.* [34]. They had achieved 17 features that have the ability to classify the data with the accuracy of 100%. To be able to select those successful features, they used an entropy-based, an χ^2-statistics, a correlation-based (CFS), a *t*-statistics and a MIT correlation-based feature selection methods. In their approach, instead of ranking the individual features, subsets of features were scored and ranked. They checked for the validity of their results using several different classifiers like *k* – nearest neighbor algorithm (*k*-NN), C4.5, Naïve Bayes (NB), SVM and PCL (Prediction by Collective Likelihood of emerging patterns). They chose 10-fold cross-validation for their tests.

Bing *et. al.* used prostate cancer and colon cancer datasets in their research in 2004 [35]. Their approach was to use combination of some feature selection methods with clustering algorithms. These methods were ranksum test, Principle Component Analysis (PCA), clustering and t test. They used three different neural networks. In the first network, the ranksum test was used to be able to extract and select the top ranked 30 genes. In the second network, PCA was used to achieve 15 principle components. In the third network, the entire data was clustered into 50 groups consisting of genes, and then t test was used to select top 30 genes which are the significant. Here, each cluster was assumed to belong to the same pathway, so they hoped to be able to prefilter the set of genes by eliminating the highly correlated ones. Then the averages of the results were taken to set the output. They made the verification of their results using 3-fold, 10-fold, leave-one-out cross-validation (LOOCV). Using this technique with those two datasets, they achieved the best results that had ever been obtained up to 2004. With this approach, they were able to select 30 features within the colon cancer dataset classifying the data with the accuracy of 91.4% and 30 features within the prostate cancer dataset classifying the data with the accuracy of 97.06%.

4.2 Experiments

Cross validation is a confirmation technique in which the data is partitioned in to subsets. The algorithm is performed on the initial subset which is called the training set. And the other testing subsets are used to validate the initial analysis. In n-fold cross validation, the data is partitioned in to n subsets and the one of them is used as a test, while there are n-1 training sets. This process is repeated n times to be able to let all n subsets to become training set. Then as a result the average of each fold is taken to give a single result.

In our work, we used LIBSVM and we only changed one of the constant parameters of LIBSVM which is the cost parameter C. That parameter is a user specific parameter. The default value for this parameter was 1 and replaced the value with 100. The parameter gamma which sets the kernel function used in the SVM is 1/k by default, where k represents the number of training sets. Thus, to be able to validate a result with n features, k is assigned to n.

4.2.1 Experiments with Five-Fold Cross-Validation

We ran the algorithm with five-fold cross validation, and we got relevant features classifying the data with the accuracy of 100%. Our algorithm is run for 500 generations for each three datasets.

Below the results we obtained from each of there datasets compared with some other results in the literature and explained separately:

	COLON CANCER DATASET	
	(having 2000 features originally)	
	ACCURACY	# of FEATURES
Our Features	100%	4
The Best Result so far	98.38%*	12*
Other Previous Results	91.4%**	30**

*Results from Küçükural *et. al.* [9]

**Results from Bing *et. al.* [35]

Table 4.1 Classification Accuracy of the Results from Colon Cancer Dataset Compared with the Other Results in the Literature

The table above presents the comparison of our results with other results in the literature. For the colon tumor dataset, the first subset of features that we obtained consists of 15 features, those feature were able to classify the entire data with the accuracy of 100%. This subset is achieved in the 100. generation. At the end of 500 generations of runs, we were able to classify the entire colon tumor dataset using only 4 features with the accuracy of 100%. We got this solution at the generation of 215. In the literature highest classification accuracy was again obtained by the previous work of our group. The classification accuracy of that approach was 98.38% using 12 features. Other result which has the classification accuracy of 91.4% using 30 features, was obtained by Bing *et. al.* in 2004. Considering this information, we can say that our approach is the first one to converge to the accuracy of 100% and for this accuracy just 4 features were used.

	PROSTATE CANCER DATASET	
	(having 12600 features originally)	
	ACCURACY	**# of FEATURES**
Our Features	100%	3
The Best Result so far	96.07%*	19*
Other Previous Results	97.06%**	30**

*Results from Küçükural *et. al.* [9]

**Results from Bing *et. al.* [35]

Table 4.2 Classification Accuracy of the Results from Prostate Cancer Dataset Compared with the Other Results in the Literature

This table shows the results we obtained from the prostate cancer dataset and comparison of this result with the other approaches in the literature. In this dataset, first result with the accuracy of 100% using only 17 features was obtained at the generation of 110. Our algorithm converged to again the accuracy of 100% using only 3 features out of 12600 features that exist in the dataset. This result had taken in the generation 216. In this dataset, the best previous best result again belongs to our group's previous work. The results of that approach were 96.07% using 19 features. Another approach by Bing *et. al.*, was able to classify the data using 30 features with the accuracy of 97.06%. So, our approach is the first one to be able to achieve the classification accuracy of 100% with a feature subset. In addition we used the subset consisting of only 3 features to get this accuracy.

	OVARIAN CANCER DATASET (having 15154 features originally)	
	ACCURACY	**# of FEATURES**
Our Features	100%	3
The Best Result so far	100%*	12*
Other Previous Results	100%***	17***

*Results from Küçükural *et. al.* [9]

***Results from Liu *et. al.* [34]

Table 4.3 Classification Accuracy of the Results from Ovarian Cancer Dataset Compared with the Other Results in the Literature

Table 4.3 summarizes our results taken from the ovarian cancer dataset. The ovarian cancer dataset, we found 29 features to classify the data. This was the first result which gave the accuracy of 100% using this dataset. Finally we achieved the classification accuracy of 100% using just 3 features at the generation of 204. Our group's previous approach which had given the best previous result, could also achieve the classification accuracy of 100%. But they used a larger of features which is 12, not to lower the accuracy. Other approaches could obtain 17 features that classify the data with the accuracy of 100%. So in the literature there exist approaches that converge to some number of feature subsets and give the accuracy of 100%, but our approach reached this accuracy using very few numbers of features.

4.2.2 An Experiment with Leave-One-Out Cross-Validation

Increasing the number of folds greatly increases the computational overhead. On the other hand, any tool should confirm the same results with the same data. As mentioned in the previous chapter, originally, we used a population having maximum 30 features for all of the datasets, and the constant parameter *feature repeat* (see chapter 3.2) was assigned 2; after switching to LOOCV, we had to fall back to maximum number of 20 features, with an initial repeat factor of 1 and we ran the algorithm for 1000 generations. We only tried one dataset for the test and we chose the prostate cancer dataset.

Our results for the prostate cancer dataset are shown in the following graph (Figure 4.1). The horizontal axis is the number of features, and the vertical axis is the number of generations a solution is found. The dots represent the solutions achieved in the corresponding generation using the corresponding number of feature with the accuracy of 100%.

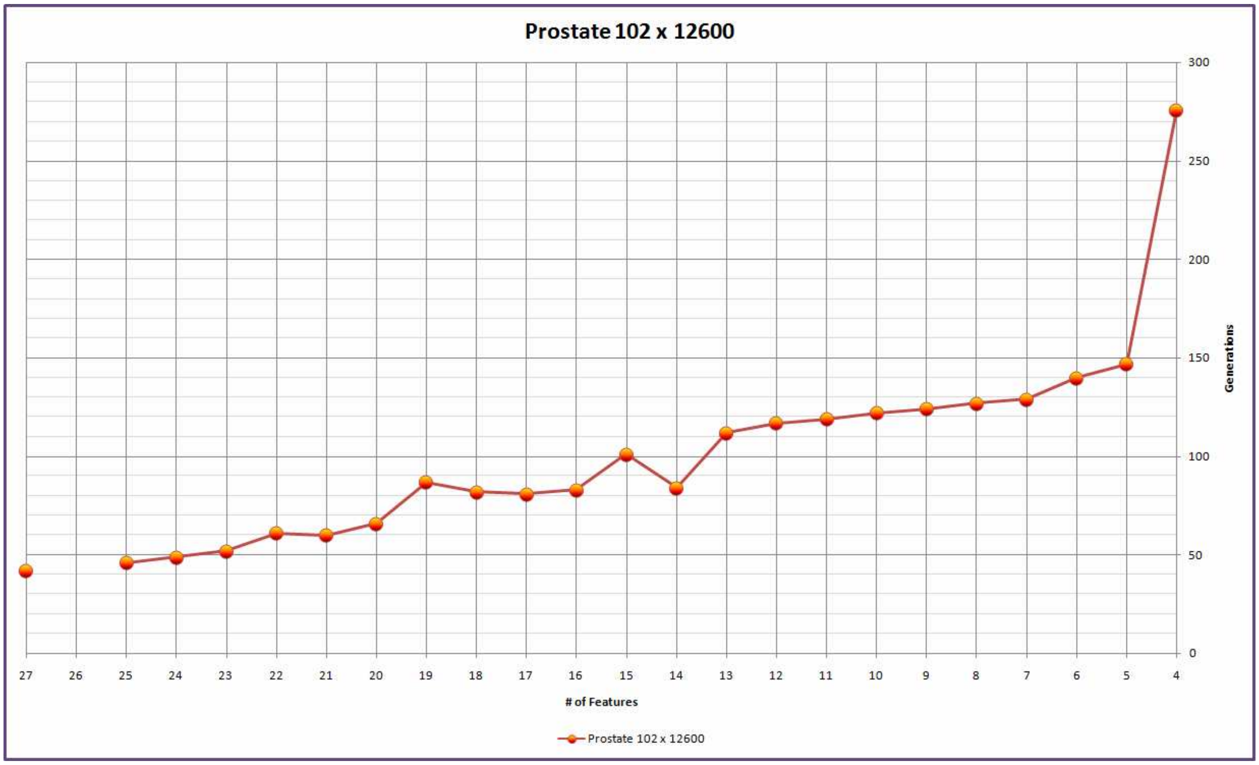


Figure 4.1 The pattern of the results

As it can be seen from the graph above, we were able to achieve at least 4 features with the validation accuracy 100%. Figure 4.1 shows the curve of the converging results through generations. The algorithm finds 27 features having the accuracy of 100% at the generation of 42 and finally it converges up to only 4 features at the generation of 276 with the accuracy of 100%.

4.3 Independent Tests to Verify Our Results

For all of the results aforesaid, we made independent tests using eighteen different classifiers which are Support Vector Classifier, Bayes-Normal-1, Bayes-Normal-2, Fisher, Polynomial Classifier, Decision Tree, Quadr, Linear Perceptron, PC Bayes-Normal-1, KL Bayes-Normal-1, Random Neural Net, K-NN Classifier, Parzen Classifier, Parzen Classifier, Bayes-Normal-U, Subspace Classifier, Scaled Nearest Mean and Nearest Mean. So that were able to confirm that we classified the dataset using our subsets of features with the accuracy of 100%.

5 CONCLUSION AND DISCUSSION

For diagnostic purposes, it is crucial to be able to identify the genes that are included in a disease pathway. On the other hand it is not possible to make a heuristic search within a microarray dataset consisting of real life disease data, since these datasets contain huge numbers of features and therefore the search space is too large. To deal with such large datasets, eliminating this data and filtering out the signature genes for related disease (selecting the features) is very important.

According to the information above, in this thesis our aim was to select the minimum set of features (genes) that has the ability to distinguish the diseased and control patients, out of three different real life cancer datasets. All of these three datasets consist of massive amount of n gene (feature) expression information.

To be able to handle this problem, in our work, we preferred to use Genetic Algorithms in combination with SVM. In the literature there are several researches using GAs in combination with various classification methods. In most of those researches, a fix number of features are used within the GA. Searching the space with different combinations of features was the only responsibility of GA. Every individual consisting of these combinations of features (genes) were only capable of retaining the fixed number of features. The classification accuracies were kept while the GA is running. After that according to these accuracies, the number of the features was tried to be reduced.

The previously proposed method by our group for this problem, set another point of view for reducing the number of features. In that approach, any gene could have a personal fitness value during parent generation step. By using the information obtained from the previous generations, the fitter individuals were generated for new population of every new generation. And in every specific number of generations, the whole population was destroyed and a totally new population was constituted according to the survival probabilities of the genes that were calculated and stored within the previous generations. During this reconstruction, the number of the features within an individual was hoped to reduce. With this approach, it became possible to converge faster then other approaches in the literature.

On the other hand, we were concerned about the importance of the unchecked genes in the very first step of the algorithm. During initialization of GA, 30 genes were selected for each individual and a population of 100 individuals was created. Therefore, genes that were selected randomly in the very beginning of the algorithm were being favored because they had the chance to increase their survival probability. After this initialization roulette wheel selection was getting stared and unfairly favored genes had higher probability to be chosen. Thus, probably the search was occurring around almost those same features. The genetic operator mutation is never enough to compensate this huge amount of information loss.

Our approach to handle this problem consists of two phases;

- We constituted a control mechanism responsible to the check that every feature in the whole database should occur in the initial population twice, while same features are not allowed to be assigned to the same parent.

- In the very beginning of the algorithm a specific number of initial runs are made, to be able to give equal chance to any gene in the database, to exhibit its classification accuracy. In those initial runs, roulette wheel selection was not used, therefore the genes with higher classification accuracies was not favored. This inhibits the algorithm to concentrate on the features that are not relevant but just luckily exist in combination with significant features within one parent. After these completely random runs, every feature will have the chance to gather its classification accuracy in combination with several random features. This will lead the main GA truly.

Another major change in the idea of our previous algorithm was in the part that whole population is destroyed in every certain number of generations.

- We decided that, this affects the flow of the algorithm negatively that it causes loss of some information gained in the previous group of generations. Thus, we decided to keep the progress of the algorithm steady. The number of features is reduced with the merges in the child. These merges occur when the two same features are assigned to one child.

In addition to the differences that are mention above, we also used a different cross-over technique in our algorithm, to be able to increase the variations in the search space.

Applying our algorithm to all three databases that we used for our tests, we achieved by far the best results in the literature. In each dataset, we obtained the minimum subset of significant features (that had ever been achieved) with the classification accuracy of 100%.

5.1 Biological Relevancy of Our Results

Our results achieved from the colon tumor dataset are shown in Table 5.1. (Related colon tumor dataset can be obtained from [38])

Attribute 186	**homosapiens thyroid receptor interactor (TRIP 1) mRNA**
Attribute 493	**Myosin Heavy Chain, Non-Muscle (gallus gallus)**
Attribute 1110	**human heat shock, e.coli homologue mRNA**
Attribute 1740	**human semenogelin II (SEMGII) gene**

Table 5.1 Genes Achieved from Colon Tumor Dataset

One of the genes that we obtained is called *homosapiens thyroid receptor interactor (TRIP 1) mRNA*. TRIP 1 (the TGFβ protein) is a cytoplasmic WD-domain protein. Some of the TGFβ responsive pathways are enhanced by the over expression of TRIP 1 [39,40]. TGFβ has responsibility in cell proliferation and differentiation. In human cancers, generally changes in the signaling of TGFβ (like any mutations or deletions within the signaling pathway) are observed. Therefore, for the diagnostics of cancer TGFβ and the members in its the signaling pathway are characteristic [41]. The type II TGFβ which is abbreviated as TβRII, is a cell serine/theorine kinase receptor (STRAP) [42,43]. In colon cancer patients a loss of TβRII is highly observed [44,45].

Myosin Heavy Chain, Non-Muscle (gallus gallus) is one of our four results which classified the entire colon tumor data with the accuracy of 100%. In the literature there a number of approaches which indicate this gene as one of the most relevant genes that is involved in the pathway of colon cancer [46,47,48]. It was also experimentally ascertained that the NMHC is a target for the protein which is encoded by mts-1 gene. Mts-1 gene is known as a metastasis-related gene [49].

Human Heat Shock, e.coli homologue mRNA and Human *Semenogelin II (SEMGII) gene* are the other genes that we obtained from colon tumor dataset.

Our results achieved from the prostate cancer dataset are shown in Table 5.2. (Related prostate cancer dataset can be obtained from [50])

37639_at	**hepsin (transmembrane protease, serine 1)**
41504_s_at	**v-maf musculoaponeurotic fibrosarcoma oncogene homolog (avian)**
1991_s_at	**mitogen-activated protein kinase-activated protein kinase3**

Table 5.2 Genes Achieved from Colon Tumor Dataset

One of our genes that we had achieved from the prostate cancer data set is called *hepsin (transmembrane protease, serine 1)* and this gene is abbreviated as HPN. HPN is known to be related with the prostate cancer because transmembrane cell surface serum is encoded by HPN and in prostates cancer transmembrane cell serum protease is over expressed [51].

Another gene that we achieved from prostate cancer dataset, as a gene that has the ability to classify the diseased and control patients is called *mitogen-activated protein kinase-activated protein kinase3 (MAPKAPK3).* MAPKAPK3 is a member of serine/threonine kinases. It is known that it is phosphorylated by a MAP kinase family (like p38, JNK, ERK). ERK which activates MAPKAPK3, is also known to be activated in tissues of several cancer types including prostate cancer [52].

The other gene we obtained from this dataset is called *MAF (v-maf musculoaponeurotic fibrosarcoma oncogene homolog)* which is an oncogene that is known to cause cancer.

6 REFERENCES

[1] Bing Lui, Qinghua Cui, Tianzi Jiang and Songde Ma, “A Combinational Feature Selection and Ensemble Nueral Network Method for Classification of Gene Expression Data”

[2] Berrar, Daniel P., “Practical Approach to Microarray Data Analysis”, *Secaucus, NJ, USA: Kluwer Academic Publishers, 2002.*

[3] Dubont Sj, Fridlyand J, Speed T , “Comparison of Discrimination Methods for The Classification of Tumors Using Gene Expression Data.”, *J Am Stat Assoc 2002, 97: 77-87*

[4] Jaeger J, Sengupta R, Ruzzo WL, “Improved Gene Selection for Classification of Mircoarrays.” *Pac Symp Biocomput 2003: 53-64*

[5] Micheal L. Raymer, William F. Punch, Erik D. Goodman, Leslie A. Kuhn, and Anil K. Jain, “Dimensionality Reduction Using Genetic Algorithms”

[6] A. K. Jain and D. Zongker, “Feature Selection: Evoluation, Application and Small Sample Performance”, *IEEE Trans. Pattern Anal. Machine Intel., Vol. 19, pp. 153-158, Feb. 1997*

[7] F. J. Ferri, P. Pudil, M. Hatef and J. Kittler, “Comparative Study of Techniques for

Large-Scale Features Selection in Pattern Recognition in Practice IV, Multiple Paradigms, Comparative Studies and Hibrit Syst." *Amsterdam:Elsevier, 1994, pp. 403-413*

[8] Holger Fröhlich, Olivier Chapelle, "Feature Selection for Support Vector Machines by Means of Genetic Algorithms", *Department Empiricial Inferemce, Max-Planck-Institude of Biological Cybernetics, Tübingen, Germany*

[9] Alper Küçükural, Reyyan Yeniterzi, Süveyda Yeniterzi, O. Uğur Sezerman, "Evolutionary Selection of Minimum Number of Features for Classification of Gene Expression Data Using Genetic Algorithms"

[10] B. Schölkopf, A. J. Smola, "Learning with Kernels", *MIT Press; CambridgeMA 2002*

[11] V.N. Vapnik, "Statistical Learning Theory", *New York 1998*

[12] J.H.Holland, "Adaptation In Natural and Artificial Systems." *The University of Michigan Press, Ann Arbour (1975)*

[13] Goldberg, D.E. "Genetic Algorithms in Search, Optimization, and Machine Learning.", *Addison-Wesley (1989)*

[14] Colin R. Reeves, Jonathan E. Rowe. Boston, MA, "Genetic Algorithms : Principles and Perspectives: A Guide to GA Theory", *Kluwer Academic Publishers, 2003.*

[15] U. Alon, N. Barkai, D. Notterman, K. Gish, S. Ybara, D. Mack, A. Levine.
"Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Cancer Tissues Probed by Oligonucleotide Arrays", *Cell Biology, 96:6745-6750, 1999*

[16] http://www.dnareplication.info/images/dnadoublehelix.jpg

[17] http://www.elmhurst.edu/~chm/vchembook/images/582basepair.gif

[18] http://www.lshtm.ac.uk/pmbu/staff/rmcnerney/homepage/image1.gif

[19] http://en.wikipedia.org/wiki/Image:Microarray-schema.gif

[20] Petricoin EF, Ardekani AM, Hitt BA, Levine PJ, Fusaro VA, Steinberg SM, Mills GB, Simone C, Fishman DA, Kohn EC, Liotta LA, "Use of Proteomic Patterns in Serum to Identify Ovarian Cancer.", *Lancet 2002, 359:572-577*

[21] Sngh D, Febbo PG, Ross K, Jackson DG, Manola J, Ladd C, Tamayo P, Renshaw AA, D'Amico AV, Richie JP, Lander ES, Loda M, Kantoff PW, Golup TR, Sellers WR, "Gene Expression Correlates of Clinical Prostate Cancer Behavior." *Cancer Cell 2002, 1:203-209*

[22] http://www3.iptv.org/exploremore/ge/what/images/cell5.gif

[23] Charles Darwin , "Origin of the Species", *1859*

[24] http://www.geneinfo.net/images/articles/crossing_over.jpg

[25] Sankar K. Pal, Paul P. Wang, "Genetic Algorithms for Pattern Recognition", *CRC Press*

[26] Goldberg, D.E., "Genetic Algorithms in Search, Optimization and Machine Learning", *Addison-Wesley, Reading, MA, 1989.*

[27] http://en.wikipedia.org/wiki/Image:SinglePointCrossover.png

[28] http://en.wikipedia.org/wiki/Image:TwoPointCrossover.png

[29] http://en.wikipedia.org/wiki/Image:CutSpliceCrossover.png

[30] Küçükural A, Yeniterzi R, Yeniterzi S, Sezerman OU, "Evolutionary Selection of Minimum Number of Features for Classification of Gene Expression Data Using Genetic Algorithms",

[31] Makoto Matsumoto and Takuji Nishimura, "Mersenne Twister: A 623-dimensionally eduidistributed uniform pseudorandom number generator", *Keio University/Max-Planck_Insitut für Mathematik*

[32] Michael Yudell, Rob DeSalle, "The Genomic Revolution", *Washington, DC, USA: Joseph Henry Press, 2002.*

[33] Nello Cristianini and John Shawe-Taylor, "An introduction to support vector machines : and other kernel-based learning methods", *Cambridge, U.K. ; New York : Cambridge University Press, 2000*

[34] Liu H, Li J, Wong L: A, "Comparative Study on Feature Selection and Classification Methods Using Gene Expression Profiles and Proteomic Patterns." *Genome Informatics 2002, 13:51-6.*

[35] Liu, Bing; Cui, Qinghua; Jiang, Tianzi; Ma, Songde, "A Combinational Feature Selection and Ensemble Neural Network Method for Classification of Gene Expression Data", *BMC Bioinformatics 2004, 5 136*

[36] http://www.ece.eps.hw.ac.uk/Research/oceans/conferenceOSL/people/images_fabien/svm_dim2.jpg

[37] Guyon I, Elisseeff A, "An Introduction to Varibable and Feature Selection", *Journal of Machine Learning Research 3 (2003) 1157-1182*

[38] http://sdmc.lit.org.sg/GEDatasets/Datasets.html#ColonTumor

[39] http://www.ncbi.nlm.nih.gov/sites/entrez?db=Pubmed&term=7566156

[40] http://www.ncbi.nlm.nih.gov/sites/entrez?db=Pubmed&term=9813058

[41] http://www.ncbi.nlm.nih.gov/sites/entrez?Db=Pubmed&Cmd=ShowDetailView&TermToSearch=15774796&ordinalpos=12&itool=EntrezSystem2.PEntrez.Pubmed.Pubmed_ResultsPanel.Pubmed_RVDocSum

[42] http://www.ncbi.nlm.nih.gov/sites/entrez?db=Pubmed&term=8395914

[43] http://www.ncbi.nlm.nih.gov/sites/entrez?db=Pubmed&term=1310899

[44] http://www.ncbi.nlm.nih.gov/sites/entrez?db=Pubmed&term=7761852

[45] http://www.ncbi.nlm.nih.gov/sites/entrez?db=Pubmed&term=8952554

[46] YeeLeng Yap, XueWu Zhang, MT Ling, XiangHong Wang, YC Wong, Antoine Danchin, "Classification between normal and tumor tissues based on the pair-wise gene expression ratio", *BMC Cancer. 2004; 4: 72.*

[47] Li L, Darden TA, Weinberg CR, Levine AJ, Pedersen LG. "Gene assessment and sample classification for gene expression data using a genetic algorithm/k-nearest neighbor method." *Comb Chem High Throughput Screen.* 2001**;**4:727–739.

[48] Kishino H, Waddell PJ. "Correspondence analysis of genes and tissue types and finding genetic links from microarray data." *Genome Inform Ser Workshop Genome Inform.* 2000;11:83–95.

[49] Kriajevska MV, Cardenas MN, Grigorian MS, Ambartsumian NS, Georgiev GP, Lukanidin EM. “Non-muscle myosin heavy chain as a possible target for protein encoded by metastasis-related mts-1 gene.” *J Biol Chem.* 1994;269:19679–19682

[50] http://sdmc.lit.org.sg/GEDatasets/Datasets.html#Prostate

[51] Pal P, Xi H, Kaushal R, Sun G, Jin CH, Jin L, Suarez BK, Catalona WJ, Deka R,”Variants in the HEPSIN gene are associated with prostate cancer in men of European origin.”,*Hum Genet. 2006 Sep;120(2):187-92*

[52] http://www.freepatentsonline.com/20070105796.html