



Jaypee University of Information Technology
Solan (H.P.)
LEARNING RESOURCE CENTER

Acc. Num. SP07011 Call Num:

General Guidelines:

- ◆ Library books should be used with great care.
- ◆ Tearing, folding, cutting of library books or making any marks on them is not permitted and shall lead to disciplinary action.
- ◆ Any defect noticed at the time of borrowing books must be brought to the library staff immediately. Otherwise the borrower may be required to replace the book by a new copy.
- ◆ The loss of LRC book(s) must be immediately brought to the notice of the Librarian in writing.

Learning Resource Centre-JUIT



SP07011

ROUGH SET THEOROTICAL ANALYSIS ON GENETIC DATA (Gene)

Submitted in partial fulfillment
Of the requirements for the degree of
BACHELOR OF TECHNOLOGY

By

Rishab Chodha (071524)

Varuni Gang (071510)

Under the supervision of
Dr. Satish Chandra



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY
WAKNAGHAT
SOLAN, HIMACHAL PRADESH
INDIA
May- 2011

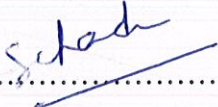
DEPARTMENT OF COMPUTER SCIENCE ENGINEERING AND INFORMATION
TECHNOLOGY

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY
WAKNAGHAT
DISTT. SOLAN, HIMACHAL PRADESH



CERTIFICATE

This is to certify that the work titled **ROUGH SET THEOROTICAL ANALYSIS ON GENETIC DATA (Gene)** submitted by **Rishab Chodha (071524)** and **Varuni Gang (071510)** in partial fulfillment for the award of degree of **B. Tech**, of Jaypee University of Information Technology, Waknaghat has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor 

Name of Supervisor **Dr. Satish Chandra**

Designation **Assistant Professor**

Date **17th May, 2011**

ACKNOWLEDGEMENTS

We hereby acknowledge with deep gratitude for the co-operation and help provided to us by all the members of this organization (**Jaypee University of Information Technology**) in preparing our final year project.

With proud privilege and profound sense of gratitude we acknowledge our indebtedness to our project guide **Dr Satish Chandra** for his valuable guidance, inputs, suggestions, consistent encouragement and co-operation.

We express our heartfelt thanks to our head of the department **Dr R.S. Chauhan** for providing us with the opportunity of doing this final year project. We extend our gratitude to all other staff members of the department of Computer Science, Jaypee University of Information Technology, Distt Solan, H.P for their numerous helps.

Rishab Chodha (071524)

Varuni Gang (071510)

Abstract

Our project is based on the use of Rough Set Theoretic analysis of gene data. The importance of rough set theory is for computing both relevance and significance of the genes. We learned how rough set theory helps to analyze the expression data and to classify the genetic data. The genetic expression data was produced by microarray processing. This genetic data was first preprocessed by use of some clustering techniques. Clustering is done for selection of some of the closely related genes, so that these can then further be analyzed for classification tasks.

To select certain number of attributes that are highly useful for classification from the remaining data, we apply rough set minimum decision rules. These rules are applied by help of RSES 2.2(Rough Set Exploration System) software which by the help of reduction algorithm such as Genetic Algorithm help us to find reducts and rules on gene expression data. RSES 2.2 was made by using Java and C++ and is a software tool that provides the means for analysis of tabular data sets with use of various methods, in particular those based on Rough Set Theory.

Thus, after calculation of rough set theoretic minimum rules we have significantly removed the inappropriate set of attributes of genetic data that were redundant and imprecise. And we are now able to classify all the genetic data with much accuracy and efficiency. The results so obtained are than used for classification of Test set. We found that a gene named Zyxin was showing 91.2% accuracy for classification of test set and 98.6% accuracy in classification of the total dataset. Our findings were than compared with the experimentally found results.

List of Figures

Figure No.	Figure Name	Page
Figure 2.1.1	Data Table	12
Figure 2.1.2	Lower and Upper approximation	13
Figure 2.1.3	Reducts	13
Figure 2.1.4	Decision Table	15
Figure 2.2.1	Optician Decision Table	16
Figure 2.2.2	Optician Reducts	18
Figure 5.2.1	Flow Diagram For K-mean Algorithm	25
Figure 6.1.1	RSES	26
Figure 6.1.2	Rules Calculation	27
Figure 6.1.3	View of Data Table Contents	29
Figure 6.1.4	View of Contents of Reducts	29
Figure 6.1.5	Information on Reduct Set	30
Figure 6.2.1	Generate Cuts	31
Figure 7.2.1	Original Data	36
Figure 7.2.2	Statistica K-mean Analysis	36
Figure 7.2.3	Selected 50 Genes	37
Figure 7.4.1	Calculated Cuts	38
Figure 7.5.1	Discretized Table	39
Figure 7.6.1	Calculated Rules	40
Figure 7.8.1	Box Plot of Zyxin	43

Table of Contents

Chapter No.	Topics	Page No.
	Certificate from the Supervisor	2
	Acknowledgement	3
	Abstract	4
	List of Figures	5
Chapter-1	Introduction	
	1.1 Genes	page 8
	1.2 Microarrays	page 8
	1.3 Problem Statement	page 9
	1.4 Objective and Scope	page 10
Chapter-2	Basic Concepts of Rough Set Theory	
	2.1 Decision Table	page 15
	2.2 Optician Decision Table	page 16
Chapter-3	History of Rough Set Theory	
Chapter- 4	Microarray Data	
	4.1 Description	page 22
Chapter- 5	Statistica	
	5.1 Cluster Analysis	page 24
	5.2 Kmean Clustering	page 24
Chapter- 6	RSES	
	6.1 RSES 2.2	page 26
	Data Discretization	
	6.2 Cuts and discretization	page 31
	ROUGH SET ALGORITHMS	
	6.3 Computing Reducts	page 32
Chapter-7	Experimental Procedure	
	7.1 Flow Diagram for the procedure	page 34
	7.2 Data Clustering	page 36
	7.3 Conversion of Data	page 37
	Data Discretization	
	7.4 Cuts calculation	page 38
	7.5 Descritization	page 39

	Rough Set Theory	
	7.6 Rules Calculation	page 40
	7.7 Reducts	page 41
	Testing	
	7.8 Evaluating with the Test set	page 43
Chapter-8	Conclusion	
	Bibliography	page 46

Chapter 1

Introduction

1.1 GENES:

A gene is a unit of heredity in a living organism. It normally resides on a stretch of DNA that codes for a type of protein. All organisms have many genes corresponding to many different biological traits, some of which are immediately visible, such as eye color or number of limbs, and some of which are not, such as blood type or increased risk for specific diseases, or the thousands of basic biochemical processes that comprise life.

In all organisms, there are two major steps separating a protein-coding gene from its protein: First, the DNA on which the gene resides must be transcribed from DNA to messenger RNA (mRNA); and, second, it must be translated from mRNA to protein. RNA-coding genes must still go through the first step, but are not translated into protein. The process of producing a biologically functional molecule of either RNA or protein is called gene expression.

1.2 MICROARRAY:

A high throughput technology that allows detection of thousands of genes simultaneously. Its Principle is base-pairing hybridization. It is a Central platform for functional genomics.

A DNA microarray consists of a solid surface, usually a microscope slide, onto which thousands of single-stranded DNA molecules have been chemically bonded. Microarray assays are based on hybridization of a single-stranded DNA labeled with a fluorescent tag to a complementary molecule attached to the chip. When each spot in a microarray is attached a unique DNA molecule, it can be used to detect presence/ absence or even concentration of a particular type of DNA molecule in test tube .The labeled nucleic acids are derived from the mRNA of a sample and so the microarray measures gene expression.

MICROARRAY GENE CHIPS

There are two types of microarray chips

- full-length cDNA chips
- oligo chips

INFORMATION DERIVABLE FROM CHIP DATA

- By detecting the quantity of fluorescent molecules attached to each spot, one can infer the relative abundance of the complementary mRNA molecules in solution.
- By observing chip data, one can infer which genes are highly expressed or not expressed, or in general the relative expression levels of all genes.
- By comparing gene expression levels under two conditions, one can infer which genes' expression levels are affected.

A/B, A-B

- By observing gene expression levels collected at different time points after a particular stimulus, one can infer how a gene's expression level changes with time

1.3 PROBLEM STATEMENT:

Microarray can be used to measure changes in expression levels in genes. There are millions of genes available for studies which are not specifically classified according to their functionalities. As the genes are highly specific in the protein it codes. There is a growing need for classification of genes for better knowledge of their specific functionality. For example in case of a data set of leukemia cells. Leukemia is a cancer of blood or bone marrow in which there is an abnormal increase of blood cells usually white blood cells. These are sub-divided into two types i.e. myeloid and lymphoblastic leukemia.

Myeloid refers to the cancerous change taking place in a type of marrow cell that normally goes on to form red blood cells while lymphoblastic refers to the cancerous change taking place in a type of marrow cell that normally goes on to form lymphocytes which are infection fighting immune system. Each of these leukemia cells require a different type of treatment depending on the blood cells they are affecting. Thus to find if a person has an abnormal increase of red blood cell or lymphocytes (i.e. is suffering from myeloid leukemia or lymphoblastic leukemia) these genes are to classified more efficiently for Drug Discovery processes. Thus to classify large amount of non specific genes is the major problem in our project.

1.4 OBJECTIVE AND SCOPE:

The objective of our project is to develop some minimum decision rules for classification of biological data with the help of rough set theory and verify its accuracy. As earlier said that it is a tedious process to classify biological data by taking into consideration their biological significance, therefore we want to generate some decision rules by using rough set theory using which we can classify the biological data easily by following these rules.

These rules must be followed by any gene taken as input and it must be classified according to these rules. Thus classification can become easy for users by following these rules. For this purpose our biological data must be converted to numerical form and then rough set concepts must be used on it.

Here we are concerned with gene data only therefore the main aim of our project is to classify a given gene sequence i.e. the order of As, Cs, Gs, and Ts by following minimum decision rules developed using rough set theory.

These minimum decision rules can be developed by taking training data i.e. some gene sequences from a database and then extracting features from it and converting the data into numerical form and finally applying rough set theory to it to get some decision rules. Once these are generated then there is no need to study the relevance of genes for classification as they can be easily classified by any user using these decision rules.

Also, these decision rules can also be developed not only for genes but for other classifications in biology. We can use rough set theory for each and every process of classification in biological data based on rough set theory which will be much user friendly and save a lot of time.

Basic Concepts of Rough Set Theory

Human knowledge about a domain is expressed by classification. Categories are features (i.e. subsets) of objects which can be worded using knowledge available in a given knowledge base. Rough set theory treats knowledge as an ability to classify perceived objects into categories. Objects belonging to the same category are considered to be indistinguishable to each other. Rough set theory has been applied mainly in data mining tasks like classification, clustering and feature selection.

Often, information on the surrounding world is

- Imprecise
- Incomplete
- Uncertain.

We should be able to process uncertain and/or incomplete information. When dealing with inexact, uncertain, or vague knowledge, the rough set theory is used. Rough set theory was introduced by Pawley in 1985. Rough sets represent a different mathematical approach to vagueness and uncertainty.

The rough set methodology is based on the premise that lowering the degree of precision in the data makes the data pattern more visible. Consider a simple example. Two acids with pKs of respectively pK 4.12 and 4.53 will, in many contexts, be perceived as so equally weak, that they are indiscernible with respect to this attribute. They are part of a rough set 'weak acids' as compared to 'strong' or 'medium' or whatever other category, relevant to the context of this classification.

Information system can be defined as

$$IS=(U,A)$$

Where, U is the universe (a finite set of objects, $U=\{x_1, x_2, \dots, x_m\}$), A is the set of attributes (features, variables) and V_a is the set of values a, called the domain of attribute a.

Consider a data set containing the results of three measurements performed for 10 objects. The results can be organized in a matrix of size 10×3 . This data can be represented in tabular format such as:

2	1	3
3	2	1
2	1	3
2	2	3
1	1	4
1	1	2
3	2	1
1	1	4
2	1	3
3	2	1

Figure. 2.1.1 Data Table

In such format, each row corresponds to an object and each column corresponds to an attribute. thus for each object there is a certain value of attribute while the value stored in it could be gene expression data or some other data i.e. Knowledge on which rough set could be applied. Thus according to the data:

$$IS = (U, A)$$

Where $U = \{x_1, x_2, x_3, x_4, x_5, x_6, \dots, x_{10}\}$, $A = \{a_1, a_2, a_3\}$

The domains of attributes are :

$$V_1 = \{1, 2, 3\}$$

$$V_2 = \{1, 2\}$$

$$V_3 = \{1, 2, 3, 4\}$$

For every set of attributes B belongs to A , an indiscernibility relation $Ind(B)$ is defined in the following way: “ two objects , X_i and X_j , are indiscernible by the set of attributes B in A , if $b(X_i)$ and $b(X_j)$ for every b belonging to B .

The notation U/A means that we are considering elementary sets of the universe U in the space A .

The rough sets approach to data analysis hinges on two basic concepts, namely lower and the upper approximations of a set, referring to:

- the elements that doubtlessly belong to the set
- the elements that possibly belong to the set
- The boundary is defined as the difference between the upper and lower approximations, contains elements which are in upper but not in lower approximation.

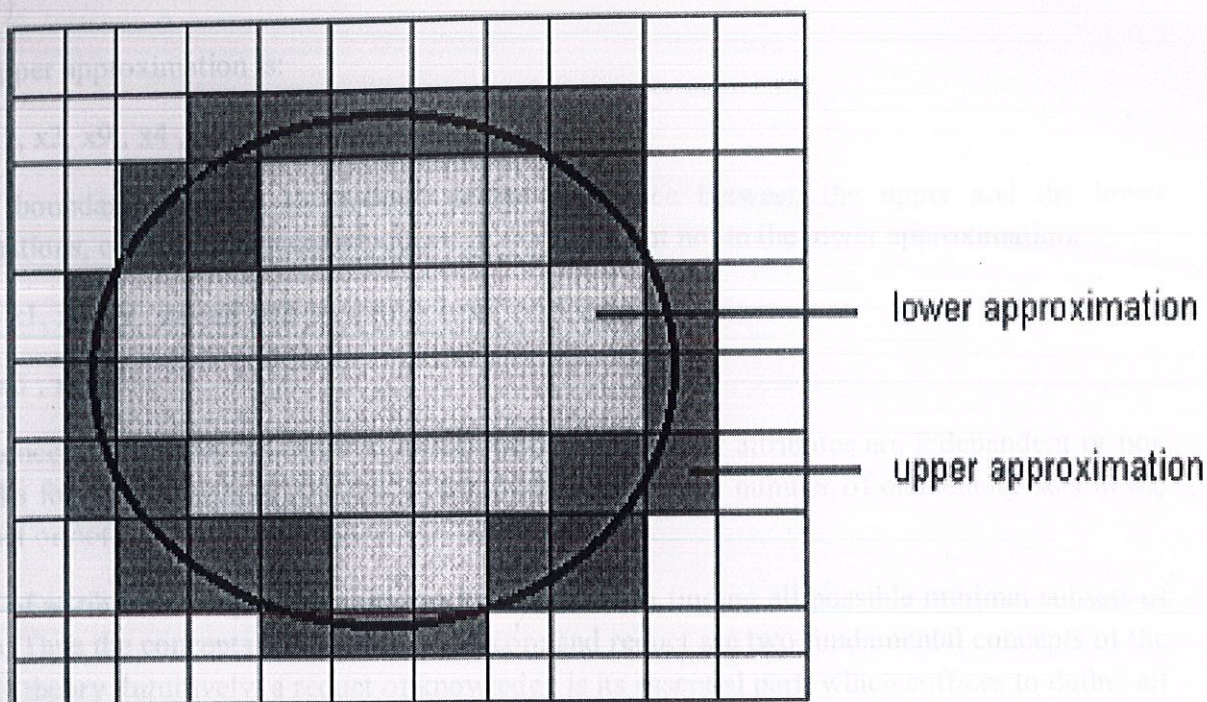


Figure 2.1.2 Lower and Upper approximation

- Let us assume that we are interested in the subset X of five objects $\{X = x_1, x_3, x_4, x_5, x_9\}$. We can distinguish this set from the whole data set in the space of three attributes. We can now calculate the lower and upper approximations of this set in the following way:
- The elementary set present in the table, which are also contained in X , are :
- $\{x_1, x_3, x_9\}, \{x_4\}$.

Table 2

U/A	a_1	a_2	a_3
$\{x_1, x_3, x_9\}$	2	1	3
$\{x_2, x_7, x_{10}\}$	3	2	1
$\{x_4\}$	2	2	3
$\{x_5, x_8\}$	1	1	4
$\{x_6\}$	1	1	2

Figure 2.1.3 Reducts

It means that the lower approximation is given by the following set of objects:

$$BX = \{x_1, x_3, x_9, x_4\}.$$

To calculate the upper approximation of the subset X, one has to find all the elementary sets in the table which have at least 1 element in common with the subset X. These are: $\{x_1, x_3, x_9\}$, $\{x_4\}$, $\{x_5, x_8\}$

So that upper approximation is:

$$BX = \{x_1, x_3, x_9, x_4, x_5, x_8\}$$

And the boundary of X in U, defined as the difference between the upper and the lower approximations, contains elements which are in the upper but not in the lower approximation:

$$\begin{aligned} BNX &= \{x_1, x_3, x_9, x_4, x_5, x_8\} - \{x_1, x_3, x_9, x_4\} \\ &= \{x_5, x_8\}. \end{aligned}$$

Independence of attributes: In order to check, whether the set of attributes are independent or not, one checks for every attribute whether its removal increases the number of elementary sets in the original set or not.

If the set of attributes is dependent, one can be interested in finding all possible minimal subsets of attributes. Thus the concepts for calculation of core and reduct are two fundamental concepts of the rough sets theory. Intuitively, a reduct of knowledge is its essential part, which suffices to define all basic concepts occurring in the considered knowledge, whereas the core is in a certain sense its most important part.

To compute reducts and core, the discernibility matrix is used. The discernibility matrix has the dimension $n \times n$, where n denotes the number of elementary sets and its elements are defined as the set of all attributes which discern elementary sets $[x]_i$ and $[x]_j$.

2.1 DECISION TABLES:

A decision table is a kind of prescription, which specifies what decisions (actions) should be undertaken when some conditions are satisfied. Most decision problems can be formulated employing decision table formalism; therefore, this tool is particularly useful in decision making.

U		a	b	c	d	e
1		1	0	2	2	0
2		0	1	1	1	2
3		2	0	0	1	1
4		1	1	0	2	2
5		1	0	2	0	1
6		2	2	0	1	1
7		2	1	1	1	2
8		0	1	1	0	1

Table 1

Figure 2.1.4 Decision Table

In this table: a , b , c and d are condition attributes and e is a decision attribute.

2.2 Opticians Decision Table:

DECISION MAKING: Used for finding minimum decision rule. Suppose we have a decision table with 24 elements

U	a	b	c	d	e
1	1	1	2	2	1
2	1	2	2	2	1
3	2	1	2	2	1
4	3	1	2	2	1
—	—	—	—	—	—
5	1	1	1	2	2
6	1	2	1	2	2
7	2	1	1	2	2
8	2	2	1	2	2
9	3	2	1	2	2
—	—	—	—	—	—
10	1	1	1	1	3
11	1	1	2	1	3
12	1	2	1	1	3
13	1	2	2	1	3

Figure 2.2.1 Optician Decision Table

Here we have a decision table for which we have to find out the decision rules, basically we have to find out the core and the reducts on the basis of which our classification will be done. This can be done by removing the inconsistency and redundancy in the data.

Here in this table a,b,c,d,e are the attributes and these 1,224 are the objects.

Now let's delete the attribute one by one and see which all rules are showing inconsistency. Let's take e attribute as a result.

When we removed attribute a the inconsistent rules are :

- $b2c2d2 = e1(\text{rule } 2)$ and $b2c2d2 = e3(\text{rule } 18 \text{ \& } 24)$
- $b1c1d2 = e2(\text{rule } 5)$ and $b1c1d2 = e3(\text{rule } 20)$

When we removed attribute b the inconsistent rules are :

- $a2c2d2 = e1(\text{rule } 3)$ and $a2c2d2 = e3(\text{rule } 18)$
- $a3c2d2 = e1(\text{rule } 4)$ and $a3c2d2 = e3(\text{rule } 24)$
- $a3c1d2 = e2(\text{rule } 9)$ and $a3c1d2 = e3(\text{rule } 20)$

When we removed attribute c the inconsistent rules are :

- $a1b1d2 = e1(\text{rule } 1)$ and $a1b1d2 = e2(\text{rule } 5)$
- $a1b2d2 = e1(\text{rule } 2)$ and $a1b2d2 = e3(\text{rule } 13)$

When we removed attribute d the inconsistent rules are :

- $a1b1c2 = e1(\text{rule } 1)$ and $a1b1c2 = e3(\text{rule } 11)$
- $a1b2c1 = e2(\text{rule } 6)$ and $a1b2c1 = e3(\text{rule } 12)$
- $a1b1c1 = e2(\text{rule } 5)$ and $a1b1c1 = e3(\text{rule } 10)$

We find that all the attributes are important and no attribute can be removed.

So, we have to compute core values of each decision rule in the decision table i.e. find all those condition attribute values in the decision rule which make the decision.

For example in the 1st decision rule $a1b1c2d2 = e1$ values c2 and d2 are core values because the rules

$b1c2d2 = e1$ and $a1c2d2 = e1$ are true whereas the rules $a1b1d2 = e1$ and $a1b1c2 = e1$ are false

The 1st rule $a1b1c2d2 = e1$ has two reducts $a1c2d2 = e1$ and

$b1c2d2 = e1$, since both decision rules are true.

In the similar manner find out all the reducts and for all the decision rules.

If the particular rule has 1 reduct then it will be represented by the rule number but if a particular rule has more than 1 reducts then they are numbered by adding ' and '

Eg: rule 1 has 2 reducts it will be represented by 1 and 1'

Like this we will calculate all the reducts and will represent in the table.

Now we will see the reducts which are giving the same results. We will find out all those and we will combine them and will represent as one.

This is shown in the tables below:

U	a	b	c	d	e
1	X	1	2	2	1
1'	1	X	2	2	1
—	—	—	—	—	—
2	1	X	2	2	1
3	X	1	2	2	1
4	X	1	2	2	1
—	—	—	—	—	—
5	1	X	1	2	2
—	—	—	—	—	—
6	1	X	1	2	2
6'	X	2	1	2	2
—	—	—	—	—	—
7	2	X	1	2	2

Figure 2.2.2 Optician Reducts

U	a	b	c	d	e
1	-	-	2	2	1
2	1	-	2	2	1
3	-	1	2	2	1
4	-	1	2	2	1
—	—	—	—	—	—
5	-	-	1	2	2
6	-	-	1	2	2
7	2	-	1	2	2
8	-	-	1	2	2
9	-	2	1	2	2
—	—	—	—	—	—
10	-	-	-	1	3

Now minimal decision rules are made by making the elementary sets. These are as follow:

U	a	b	c	d	e
1',2	1	X	2	2	1
1,3,4	X	1	2	2	1
—	—	—	—	—	—
5,6	1	X	1	2	2
7,8	2	X	1	2	2
6',8',9	X	2	1	2	2
—	—	—	—	—	—
10- 17,19'					
21,22,23	X	X	X	1	3
17',18	2	2	2	X	3
19,20	3	2	2	X	3
23',24	3	2	2	X	3

$$a1c2d2 = e1$$

$$b1c2d2 = e1$$

$$a1c1d2 = e2$$

$$a2c1d2 = e2$$

$$b2c1d2 = e2$$

$$d1 = e3$$

$$a2b2c2 = e3$$

$$a3b1c1 = e3$$

$$a3b2c2 = e3$$

Now name them as one i.e represent the elementary sets as one

U	a	b	c	d	e
1	1	X	2	2	1
2	X	1	2	2	1
—	—	—	—	—	—
3	1	X	1	2	2
4	2	X	1	2	2
5	X	2	1	2	2
—	—	—	—	—	—
6	X	X	X	1	3
7	2	2	2	X	3
8	3	2	2	X	3
9	3	2	2	X	3

Crosses in the table denote “don’t care” values of attributes. What we have obtained finally is the minimal set of decision rules (minimal decision algorithms) which is equivalent to the original table as far as the decisions are concerned. That means that is the simplified table only the minimal set of conditions, necessary to make decisions specified in the table, are included.

Now the final step is to make apply the Decision algorithm and we will get the following result

$$(a1 \vee b1)c2d2 = e1$$

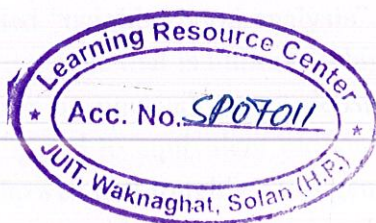
$$(a1 \vee a2 \vee b2)c1d2 = e2$$

$$d1 \vee (a3b2c1) \vee ((a2 \vee a3)b2c2) = e3$$

HISTORY OF ROUGH SET THEORY FOR THE PURPOSE OF GENE SELECTION

Recent years, rough sets theory has been used in gene selection task by some researchers. Evolutionary rough feature selection has been employed on three gene expression datasets by M. Banerjee, S. Mitra and H. Banka .They have applied an evolutionary rough feature selection algorithm for classification of microarray gene expression patterns. Rough set theory is employed to generate reducts, which represent the minimal sets of non redundant features capable of discerning between all objects, in multi-objective framework. And then they have shown the effectiveness of their algorithm on the cancer datasets. They by this algorithm selected 10 genes from each data set and high classification accuracies were obtained.

A positive region based reduct algorithm was also developed by B.F. Momin,S. Mitra and R. Datta Gupta . Identification of gene subsets responsible for discerning between available samples of gene microarray data is an important task in Bioinformatics. They have presented an algorithm for generating reducts from gene microarray data. It proceeds by processing gene expression data, discretization of real value attributes into categorical followed by positive region based approach for reduct generation. They have also discussed different approaches for reduct generation. They have found that more than 90% of redundant genes are eliminated.



Microarray Data

4.1 Description:

Golub et al. set out to develop a systematic approach to cancer classification based on the simultaneous expression monitoring of thousands of genes using DNA microarrays. It has been suggested that such microarrays could provide a tool for cancer classification. Microarray studies to date, however, have primarily been descriptive rather than analytical and have focused on cell culture rather than primary patient material, in which genetic noise might obscure an underlying reproducible expression pattern.

They began with class prediction: How could one use an initial collection of samples belonging to known classes (such as AML and ALL) to create a “class predictor” to classify new, unknown samples? They developed an analytical method and first tested it on distinctions that are easily made at the morphological level, such as distinguishing normal kidney from renal cell carcinoma. They then turned to the more challenging problem of distinguishing acute leukemia, whose appearance is highly similar. Their initial leukemia data set consisted of 8 bone marrow samples (27 ALL, 11 AML) obtained from acute leukemia patients at the time of diagnosis.

RNA prepared from bone marrow mononuclear cells was hybridized to high-density oligonucleotide microarrays, produced by Affymetrix and containing probes for 7129 human genes. For each gene, we obtained a quantitative expression level. Samples were subjected to a priori quality control standards regarding the amount of labeled RNA and the quality of the scanned microarray image. The first issue was to explore whether there were genes whose expression pattern was strongly correlated with the class distinction to be predicted. The 7129 genes were sorted by their degree of correlation.

To establish whether the observed correlations were stronger than would be expected by chance, we developed a method called “neighborhood analysis”. Briefly, one defines an “idealized expression pattern” corresponding to a gene that is uniformly high in one class and uniformly low in the other. One tests whether there is an unusually high density of genes “nearby” (that is, similar to) this idealized pattern, as compared to equivalent random patterns. For the 38 acute leukemia samples, neighborhood analysis showed that roughly 1100 genes were more highly correlated with the AML-ALL class distinction than would be expected by chance. This suggested that classification could indeed be based on expression data.

The second issue was how to use a collection of known samples to create a “class predictor” capable of assigning a new sample to one of two classes. They developed a procedure that uses a fixed subset of “informative genes” (chosen based on their correlation with the class distinction) and makes a prediction on the basis of the expression level of these genes in a new sample.

Each informative gene casts a “weighted vote” for one of the classes, with the magnitude of each vote dependent on the expression level in the new sample and the degree of that gene’s correlation with the class distinction. The votes were summed to determine the winning class, as well as a “prediction strength” (PS), which is a measure of the margin of victory that ranges from 0 to 1. The sample was assigned to the winning class if PS exceeded a predetermined threshold, and was otherwise considered uncertain. On the basis of previous analysis, we used a threshold of 0.3 .

The third issue was how to test the validity of class predictors. They used a two-step procedure. The accuracy of the predictors was first tested by cross-validation on the initial data set. Briefly, one withholds a sample, builds a predictor based only on the remaining samples, and predicts the class of the withheld sample. The process is repeated for each sample, and the cumulative error rate is calculated. One then builds a final predictor based on the initial data set and assesses its accuracy on an independent set of samples. They applied this approach to the 38 acute leukemia samples. The set of informative genes to be used in the predictor was chosen to be the 50 genes most closely correlated with AML-ALL distinction in the known samples.

The parameters of the predictor were determined by the expression levels of these 50 genes in the known samples. The predictor was then used to classify new samples, by applying it to the expression levels of these genes in the sample. The 50-gene predictors derived in cross-validation tests assigned 36 of the 38 samples as either AML or ALL and the remaining two as uncertain (PS , 0.3). All 36 predictions agreed with the patients’ clinical diagnosis. They then created a 50-gene predictor on the basis of all 38 samples and applied it to an independent collection of 34 leukemia samples.

The specimens consisted of 24 bone marrow and 10 peripheral blood samples . In total, the predictor made strong predictions for 29 of the 34 samples, and the accuracy was 100%. The success was notable because the collection included a much broader range of samples, including samples from peripheral blood rather than bone marrow, from childhood AML patients, and from different reference laboratories that used different sample preparation protocols. Overall, the prediction strengths were quite high (median PS 5 0.77 in cross-validation and 0.73 in independent test). The average prediction strength was lower for samples from one laboratory that used a very different protocol for sample preparation. This suggests that clinical implementation of such an approach should include standardization of sample preparation.

STATISTICA is a statistics and analytics software package developed by StatSoft. The software includes an array of data analysis, data management, data visualization, and data mining procedures; as well as a variety of predictive modelling, clustering, classification, and exploratory techniques. Additional techniques are available through integration with the free, open source R programming environment.

5.1 Cluster Analysis :

The term cluster analysis encompasses a number of different algorithms and methods for grouping objects of similar kind into respective categories. In other words cluster analysis is an exploratory data analysis tool which aims at sorting different objects into groups in a way that the degree of association between two objects is maximal if they belong to the same group and minimal otherwise. Cluster analysis simply discovers structures in data without explaining why they exist. The general categories of cluster analysis methods are Joining (Tree Clustering), Two-way Joining (Block Clustering), and k-Means Clustering.

5.2 k-Means Clustering :

In statistics and data mining, k-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.

An attribute clustering method can help in grouping genes based on their interdependence so as to mine meaningful patterns from the gene expression data. It can be used for gene grouping, selection and classification. The partitioning of a relational table into attribute subgroups allows a small number of attributes within or across the groups to be selected for analysis. By clustering attributes, the search dimension of a rough set algorithm is reduced. The reduction of search dimension is especially important to application of rough set theory in gene expression data because such data typically consist of a huge number of genes (attributes) and a small number of gene expression profiles (tuples).

By applying clustering algorithm such as K-mean algorithm to gene expression data, meaningful clusters of genes can be discovered. The grouping of genes based on attribute interdependence within group helps to capture different aspects of gene association patterns in each group.

Significant genes selected from each group then contain useful Information for gene expression classification and identification.

By selecting a subset of genes which have high multiple-interdependence with others within clusters, significant classification information can be obtained. Thus a small pool of selected genes can be used to build classifiers with very high classification rate. From the pool, gene expressions of different categories can be identified by the help of Rough set theory.

Given an un-annotated dataset satisfying the above assumption, we first partition it into k clusters, where each cluster comprises data-vectors with similar inherent characteristics. The data clustering task is carried out with no a priori knowledge about the intrinsic class structure—i.e. how the data is inherently partitioned into distinct clusters. In practice, the data clustering algorithm inductively derives the class information and partitions the data-set accordingly. We use the popular K-Means data clustering algorithm primarily due to its effectiveness and procedural simplicity. The net outcome of this phase is the availability of k number of data clusters, which forms the basis for subsequent discovery of symbolic rules that define the structure of the discovered clusters.

As the result of a k-means clustering analysis, we examined the means for each cluster on each dimension to assess how distinct our k clusters are. We obtained very different means for most, if not all dimensions, used in the analysis.

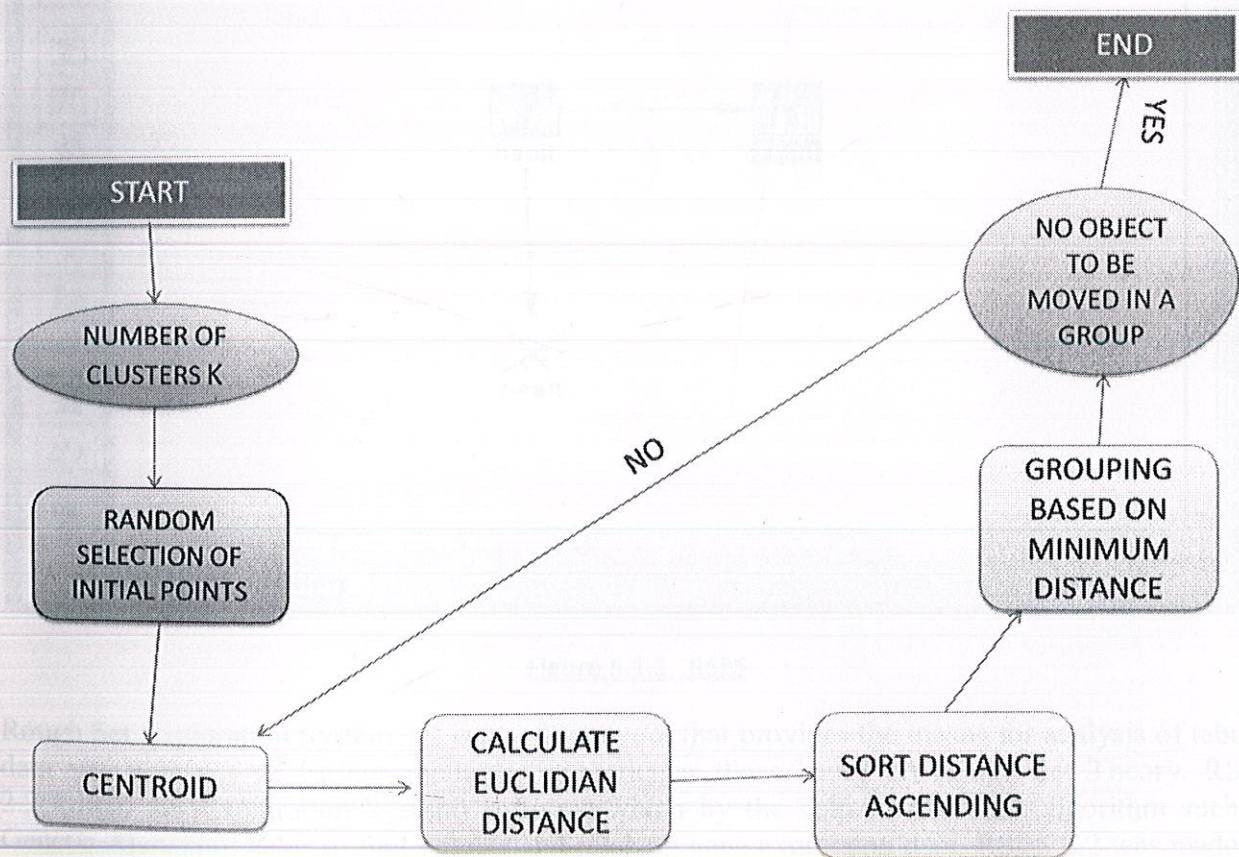


Figure 5.2.1 Flow Diagram For K-mean Algorithm

6.1 RSES 2.2(Rough Set Exploration System 2.2):

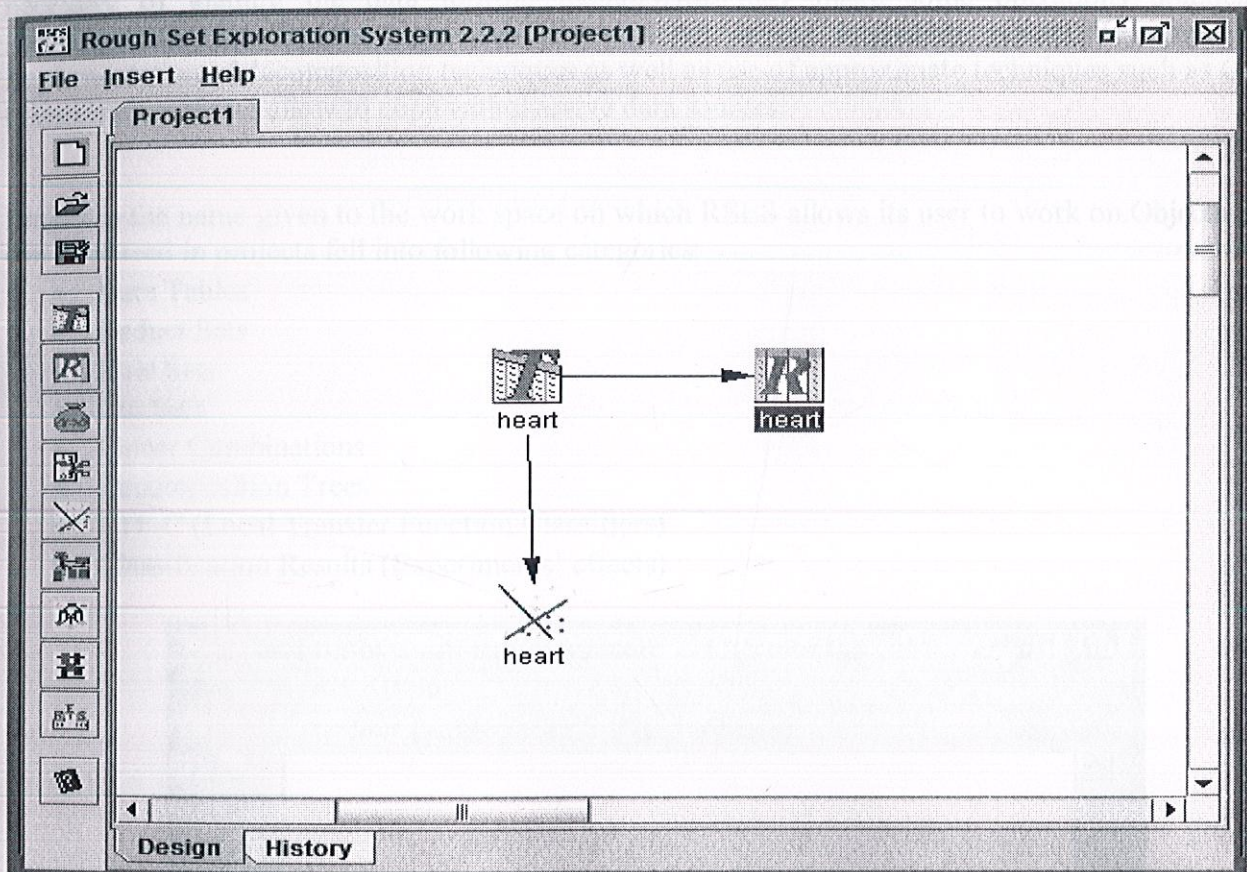


Figure 6.1.1 RSES

Rough Set Exploration System 2.2 is a software tool that provides the means for analysis of tabular data sets with use of various methods, in particular those based on Rough Set Theory. RSES 2.2(Rough Set Exploration System) software which by the help of reduction algorithm such as Genetic Algorithm helps to find reducts and rules on gene expression data. RSES 2.2 was made by using Java and C++ and is a software tool that provides the means for analysis of tabular data sets with use of various methods, in particular those based on Rough Set Theory. In general, the RSES system offers the following capabilities:

- import of data from text files,
- visualization and pre-processing of data including, among others, methods for discretization and missing value completion,
- construction and application of classifiers for both smaller and vast data sets, together with methods for classifier evaluation.

The RSES system is a software tool with an easy-to-use interface, at the same time featuring a bunch of method that make it possible to perform compound, non-trivial experiments in data exploration with use of Rough Set methods. The RSES system is capable of working with several projects at the same time. However, only one of the experiments may be active (perform computation) at any given moment.

With RSES one can explore any data that is represented as the rectangular table of reasonable size. "Reasonable" size means that for the very large data tables the significant latency caused by the necessity of loading the data to/from memory/file may make some operations practically unmanageable. So, for the huge data one can not expect immediate results. However, the implementation of decomposition techniques as well as use of approximate techniques such as GA's (genetic algorithms) allow to cope with massive data sources.

Project is the name given to the work space on which RSES allows its user to work on. Objects that can be placed in projects fell into following categories:

- Data Tables
- Reduct Sets
- Rule Sets
- Cut Sets
- Linear Combinations
- Decomposition Trees
- LTF-C (Local Transfer Function Classifiers)
- Classification Results (Experiments' effects)

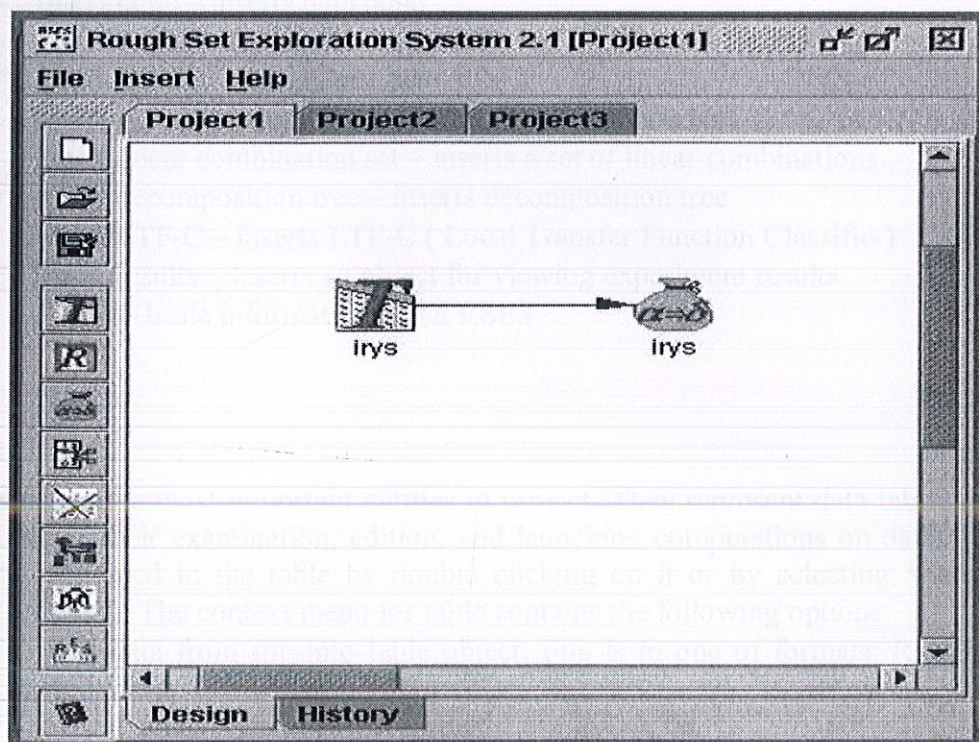
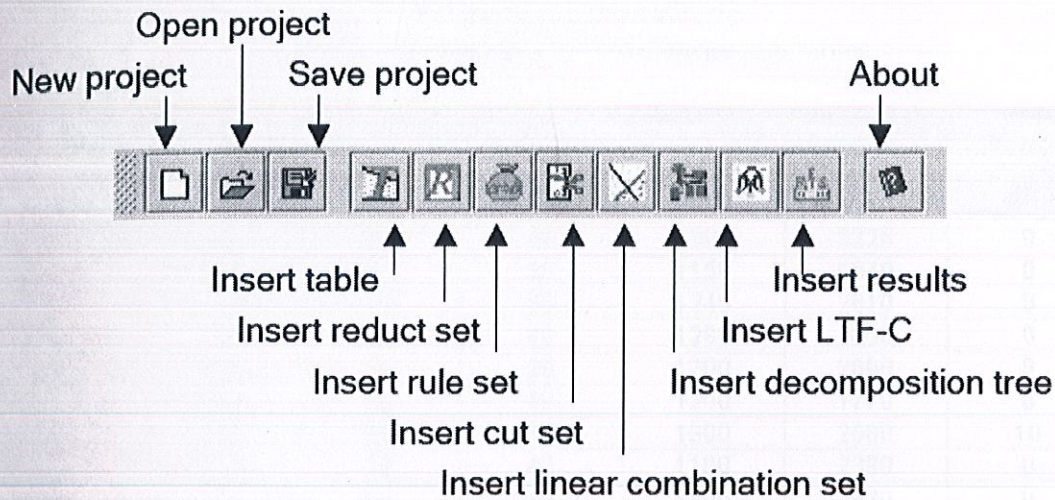


Figure 6.1.2 Rules Calculation

Dependencies between objects within project are marked by connecting such object with arrows. For example, if we calculate decision rules for some data table then the arrow originating in table and pointing at set of rules appears.



The toolbar contains buttons corresponding to selected options from main and general menus. In this way the RSES user have instant access to most common actions.

Some of the functions of the toolbar are:

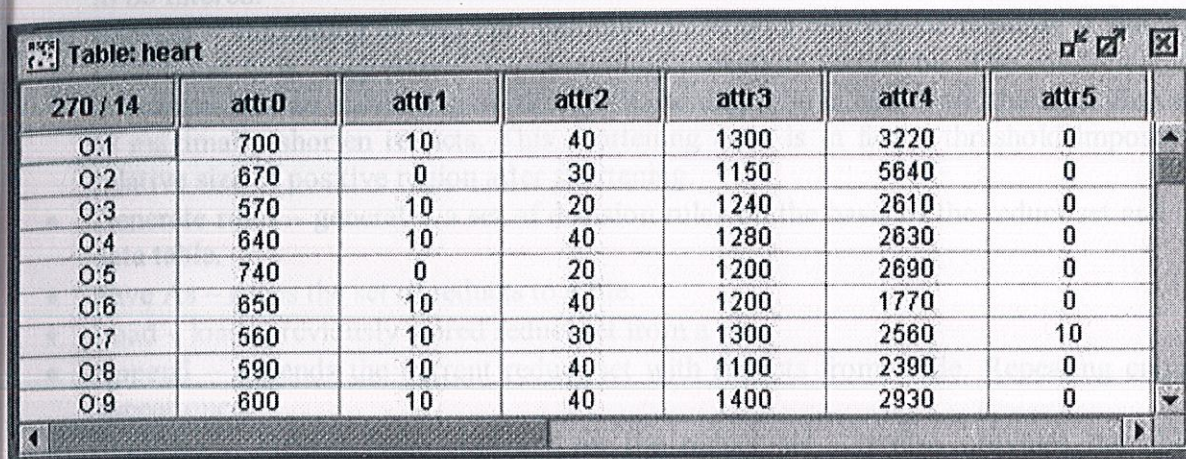
- New project – creates new project
- Open project – restores previously saved project from the disk
- Save project – saves active project to a file on disk
- Exit – terminates RSES
- Insert table – inserts data table
- Insert reduct set – inserts reduct set
- Insert rule set – inserts rule set
- Insert cut set – inserts cut and/or attribute partition set
- Insert linear combination set – inserts a set of linear combinations
- Insert decomposition tree – inserts decomposition tree
- Insert LTF-C – inserts LTF-C (Local Transfer Function Classifier)
- Insert results – inserts an object for viewing experiment results
- About – basic information about RSES



Tables are the most important entities in project. They represent data tables (tabular data sets) and allow for their examination, edition, and launching computations on data. The user can view the data contained in the table by double clicking on it or by selecting View from table object's context menu. The context menu for table contains the following options:

- Load – load data from file into table object. File is in one of formats: RSES, RSES 1.0, Rosetta, and Weka.
- Save As – save data to file in RSES format.
- View – view the contents of table .The user may Scroll, and rearrange the view window.

- Change name – change the table name (see figure 3.4). This name is saved together with data. Table name does not have to be identical with the name of file used to store the table on disk. Table name can also be altered by double-clicking on table name appearing below the icon.
- Change decision attribute – selecting the decision attribute. Selected attribute is moved to the end of table (becomes the last attribute).
- Remove – removes table (after separate confirmation).
- Split in Two – randomly splits table into two disjoint sub tables.

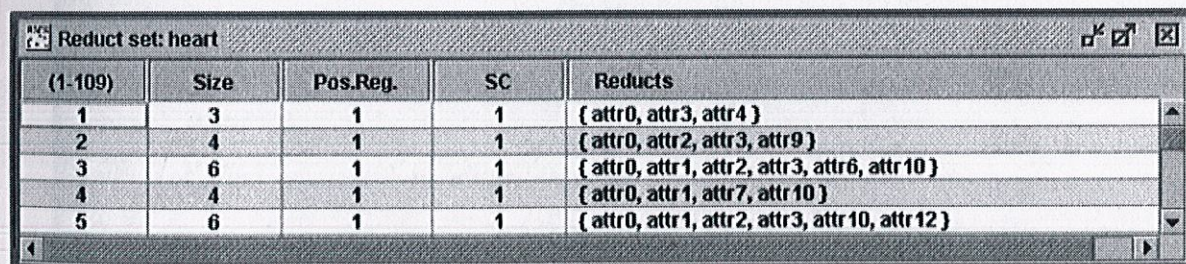


270 / 14	attr0	attr1	attr2	attr3	attr4	attr5
O:1	700	10	40	1300	3220	0
O:2	670	0	30	1150	5640	0
O:3	570	10	20	1240	2610	0
O:4	640	10	40	1280	2630	0
O:5	740	0	20	1200	2690	0
O:6	850	10	40	1200	1770	0
O:7	560	10	30	1300	2560	10
O:8	590	10	40	1100	2390	0
O:9	600	10	40	1400	2930	0

Figure 6.1.3 View of data table contents



Reduct for an information system is a subset of attributes which preserves all discernibility information from the information system, and none of its proper subsets has this ability.



(1-109)	Size	Pos.Reg.	SC	Reducts
1	3	1	1	{ attr0, attr3, attr4 }
2	4	1	1	{ attr0, attr2, attr3, attr9 }
3	6	1	1	{ attr0, attr1, attr2, attr3, attr6, attr10 }
4	4	1	1	{ attr0, attr1, attr7, attr10 }
5	6	1	1	{ attr0, attr1, attr2, attr3, attr10, attr12 }

Figure 6.1.4 Viewing contents of reduct set

Options in the context menu for reduct set:

- View – displays contents of the reduct set (see figure 3.15). The user can scroll and resize this window according to requirements. The reduct set view window consists of five columns. First of these columns stores the identification number, the others have the following meaning (for a single row):
 - Size – size of the reduct, number of participating attributes.
- Pos.Reg. – the positive region for the table after reduction, i.e. after removing attributes from outside the reduct.
- SC – value of the Stability Coefficient (SC) for the reduct. This value is used to determine the stability of reduct in dynamic case.
 - Reducts – reduct presented as a list of attributes.

- Change name – changes object name , the name is stored together with the contents of object in file. The name of object does not need to be identical with the name of file that is used to store it.
- The name of object can also be changed by double clicking on name tag below the icon representing object.
- Remove – removes reduct set .
- Filter – filters the reduct set. The user can remove reducts on the basis of stability coefficient (SC). Before using this option it is recommended to examine statistics for the set of reducts to be filtered.
- Shorten – shortening of reducts. The user provides a coefficient between 0 and 1, which determines how “aggressive” the shortening procedure should be. The coefficient equal to 1.0 means that no shortening occurs. If Shortening ratio is near zero, the algorithm attempts to maximally shorten reducts. This shortening ratio is in fact a threshold imposed on the relative size of positive region after shortening.
- Generate rules – generates a set of decision rules on the basis of the reduct set and selected data table.
- Save As – saves the set of reducts to a file.
- Load – loads previously stored reduct set from a file.
- Append – appends the current reduct set with reducts from a file. Repeating entries only appear once.
- Statistics – present basic statistics on the reduct set . It also provides the ability for displaying the core (intersection of all reducts).

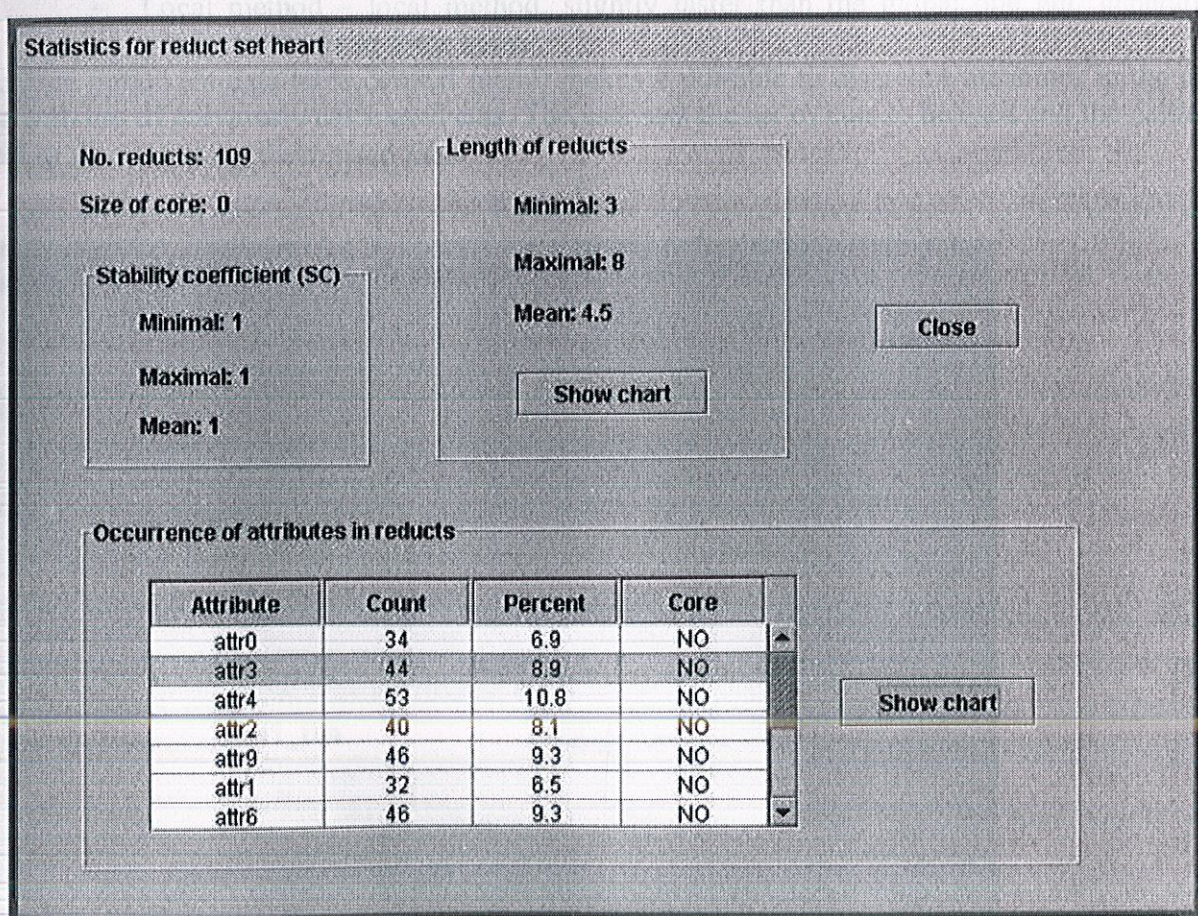


Figure 6.1.5 Information on reduct set

Data Discretization

The motivation for this phase is driven by the fact that ordinal or continuous valued attributes are proven to be rather unsuitable for the extraction of concise symbolic rules. Henceforth, the necessity to discretize continuous-valued attributes to discrete intervals—i.e. reduce the domain of values of an attribute to a small number of attribute-value ranges—where each interval can be represented by a label/token. More attractively, the data discretization phase not only reduces the complexity and volume of the data-set, but also serves as a attribute filtering mechanism, whereby attributes that are deemed to have minimum impact on the class specification can be eliminated.

6.2 Cuts and discretization:

With use of Discretize/Generate cuts from data table context menu we may generate decompositions of attribute value sets. With these descriptions, further referred to as cuts we may perform next step, i.e. discretization of numerical attributes or grouping (quantization) of nominal attributes.

The user may set several parameters that control discretization/grouping procedure:

- Method choice – choice of discretization method from:
 - Global method – global method
 - Local method – local method, slightly faster than the global one but, generating much more cuts in some cases.

Discretize option (in data table context menu) makes it possible to discretize attributes in the data table with use of previously calculated cuts. The user sets the set of cuts to be used and the name of object to store resulting discretized table.

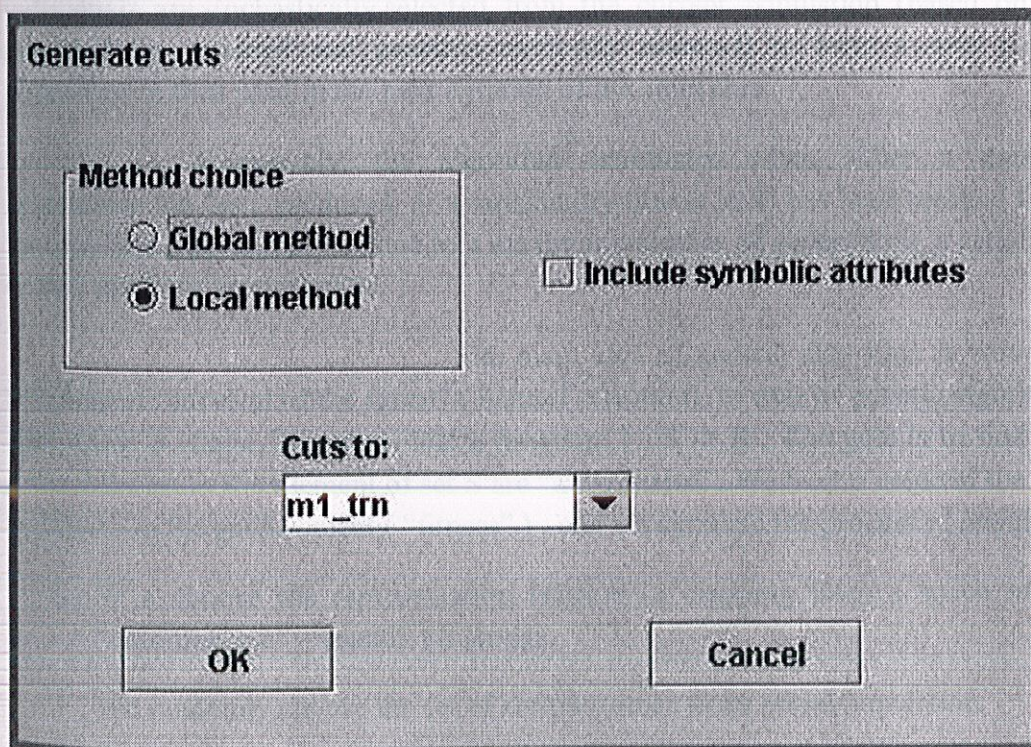


Figure 6.2.1 Generate Cuts

ROUGH SET ALGORITHMS

6.3 Computing reducts using genetic algorithms:

The time cost of the reduct set computation can be too high in case the decision table consist of too many: objects or attributes or different values of attributes. the reason is that in general the size of the reduct set can be exponential with respect to the size of the decision table and the problem of computing a minimum reduct is NP hard. One way of solving such problem is to use approximation algorithm that do not give the optimal solutions but require short computing time. Among these is the Genetic Algorithm.

In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations.

In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm.

Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

The main idea of genetic algorithm is based on the Darwinian principle of "survival of the fittest" (natural selection). In case of genetic algorithms we are given a state space S (finite, but large) and a function: $f: S \rightarrow R$. The goal is to find $x_0: f(x_0) = \max\{f(x): x \text{ belongs to } S\}$. Element of set S are "individuals". We treat a value of the function f as ability to survive in the environment ("fitness"), and we simulate the process of evolution as follows:

1. We choose the representation scheme: a mapping from a space of "individuals" into "chromosome" – usually bit strings.
2. We randomly choose the set of chromosomes as an initial population.

3. We calculate "fitness" $F(c)$ of each chromosomes as a value of $f(s(c))$, where $s(c)$ is the individual encoded by c . then we create a new population, replacing the chromosome with low fitness by those with higher fitness.
4. We randomly affect the new population by genetic operators, e.g. mutation (small random modification of chromosomes) and crossing-over (exchange of "genetic material" between some pairs of chromosomes).
5. We repeat 3-4 with new population, until a stopping criterion is satisfied.

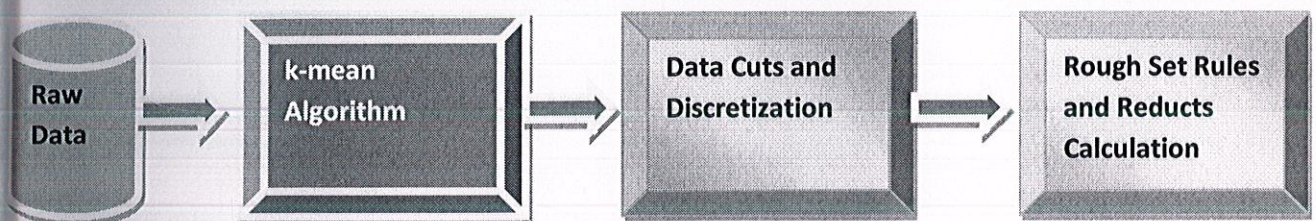
Thus resulting in the evolution of the best individual x_{max} which usually as good as the global optimum.

Experimental Procedure

Table 1. Characteristics of Dataset Used

DATASETS	No. of Samples/Patients	No. of Genes	ATTRIBUTES
Acute Leukemia	38	7129	1. ALL → 27 2. AML → 11
		7129	1. ALL → 27 2. AML → 14

7.1 Flow Diagram for the procedure:



In this paper we will present experimental results based on prediction results for Golub et al "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring". The dataset is chosen for two reasons: (1) all their data vector components being continuous-valued and (2) all the class-subsets are well-separated.

The acute leukemia dataset (<http://www.genome.wi.mit.edu/MPR>) consists of 38 samples including 27 cases of acute lymphoblastic leukemia (ALL) and 11 cases of acute myeloid leukemia (AML). The gene expression measurements were taken from high-density oligonucleotide microarrays containing 7129 genes. An independent test set of 20 ALL and 14 AML samples also exists.

Table 1. Characteristics of Dataset Used

DATASETS	No of Samples/Patients	No. of Genes	ATTRIBUTES
Acute Leukemia Training Dataset	38	7129	1. ALL => 27 2. AML => 11
Acute Leukemia Test Dataset	35	7129	1. ALL => 21 2. AML => 14

7.2 Data Clustering Using K-Means Clustering Algorithm:

Prior to clustering the actual classification information is removed from each dataset—i.e. we work with an un-annotated dataset. The K-means algorithm is used to inductively cluster the data patterns. Upon completion of the clustering process the members of each cluster are associated with their respective class label.

Use of Statistica for K-mean clustering:

1. Opening the data in excel sheet inside of Statistica.

Samples - 38 Patients																										
	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ								
1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2								
2	83	326	20	20	20	20	20	20	419	20	29	324	20	20	20	20	20	40								
3	20	469	20	20	20	20	20	20	350	24	524	167	20	176	321	29	247	20								
4	49	171	327	115	58	20	85	80	270	297	38	208	20	30	357	232	48	465								
5	20	20	20	20	20	20	20	20	20	51	20	20	20	20	2003	20	20	216								
6	126	448	73	20	30	147	191	35	20	132	20	135	85	123	1735	182	45	227								
7	370	2822	404	153	594	659	753	495	503	647	202	423	295	532	3995	378	372	722								
8	20	70	20	16000	16000	20	4958	29	7953	1065	442	45	9042	5199	1255	504	2171	580								
9	20	20	20	13390	16000	20	2666	20	4004	498	20	20	3852	1869	20	92	1752	130								
10	20	20	20	10071	5885	20	4297	20	2584	2039	20	20	2857	1365	300	114	904	20								
11	12502	12701	188	13831	16000	11329	16000	6480	9805	16000	14706	16000	16000	15496	16000	10695	16000	16000								
12	14131	10592	20	11270	16000	10299	15970	8380	7327	12453	16000	14230	15646	12086	13617	13961	7103	16000								
13	9670	10471	570	8588	16000	11295	8496	7579	12077	15706	8446	16000	16000	16000	16000	12695	9628	16000								
14	13457	16000	20	11071	16000	16000	12980	14928	16000	16000	12482	16000	16000	16000	16000	15550	13001	16000								
15	163	294	71	285	340	249	326	172	151	701	559	409	340	398	751	342	432	269								
16	20	147	52	20	45	20	220	20	20	427	375	76	47	22	494	20	257	86								
17	154	264	184	164	203	20	610	158	201	723	812	435	341	322	1474	367	331	292								
18	145	318	20	7518	7453	245	9085	322	4681	1710	269	209	3879	2113	870	561	1733	718								
19	986	912	1263	8237	10411	1070	9692	2367	3692	2644	975	1969	5114	3076	1616	2244	3481	2371								
20	808	832	740	632	370	871	1192	800	522	1174	277	405	429	941	1716	1127	1046	1007								
21	1222	1865	607	1171	1812	155	1573	794	4329	1138	310	1474	1633	1566	1693	1521	1637	1528								
22	90	20	20	142	108	75	275	20	20	319	20	273	154	48	510	209	20	160								
23	20	247	152	385	203	104	412	25	20	389	20	394	153	224	907	271	343	479								
24	241	327	104	430	286	20	879	194	303	461	38	268	307	307	437	596	351	487								
25	155	57	37	453	364	514	251	244	646	389	157	511	591	441	1013	332	639	684								
26	412	639	46	126	305	217	420	125	485	190	318	382	486	388	260	324	473	50								
27	225	2194	191	99	197	290	1559	110	178	214	103	239	221	405	1306	205	341	74								

Figure 7.2.1 Original Data

2. Calculation of K-mean Clusters with the following specifications.

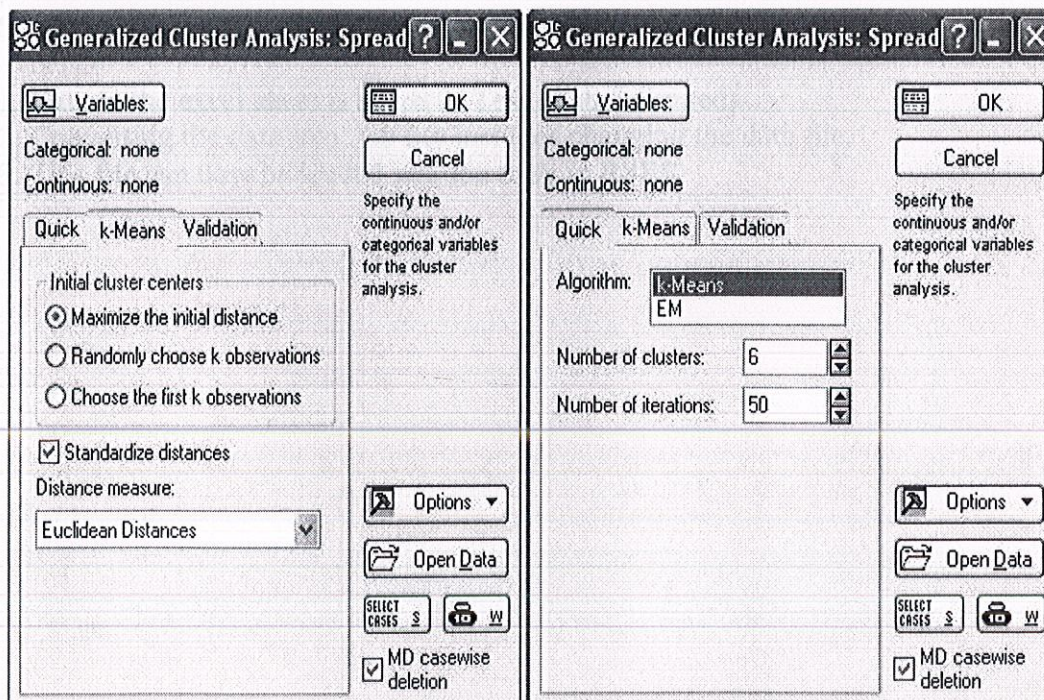


Figure 7.2.2 Statistica K-mean Analysis

3. Checking the mean for each of the 6 clusters.

- If the number of elements for the cluster is greater than 80% of the training data set => consider for further clustering.
- Else discard the data.

4. Repeat 3 until selection of closest 50 genes.

		ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL
AFFX-HSAC07/X00351_s_at (endo	AFFX-HSAC07/X00351_s_at	16287	15770	16386	13576	16301	11665	14587	16292	14150	14989	18222	9672	10468
KIAA0221 gene	U65533_s_at	-215	-483	-258	-460	-872	-354	-313	-183	-323	-807	-191	-395	-360
Globin, Beta	HG1428-HT1428_s	17962	22240	5472	33039	25829	10661	16215	1886	15551	21905	62	18798	9391
RPS3 Ribosomal protein S3	X57351_s_at	16489	10996	12769	16432	18890	16859	14353	16638	9783	25287	1182	17327	20695
LAMR1 Laminin receptor (2H5 epi	M14199_s_at	16528	17782	18496	12445	13247	17420	13180	19487	19597	16602	19533	15986	16688
Myosin, Light Chain, Alkali, Smoo	HG2815-HT2931_a	10557	4773	8029	4108	7135	6596	8542	8443	12225	8986	4415	4931	15455
Heterogeneous Nuclear Ribonucle	HG3076-HT3238_s	4094	2621	4087	1863	4251	1244	2966	3530	3217	5765	1351	2366	7284
Unknown protein gene extracted	M31520_rna1_s_a	8818	9380	8300	4333	5876	5471	5870	10147	8813	12250	6202	7906	15039
Dna-Binding Protein Hrfx2	HG3327-HT3504_s	-1198	-1322	-561	-1327	-1577	-892	-756	-670	-994	-1557	-640	-719	-834
GB DEF = 52 kD subunit of transcr	Y07595_at	-385	-67	-203	-509	-556	-280	-292	-280	-357	-568	-289	-310	-417
BETA-2-MICROGLOBULIN PRECUR	J00105_s_at	21909	18519	13909	15020	24015	16810	20133	21147	15167	24346	18367	15719	22990
mRNA fragment encoding beta-tu	V00599_s_at	16077	11421	10273	8264	11355	5868	17328	14960	6919	19825	15160	6596	13375
SERUM AMYLOID A PROTEIN PREC	X51441_s_at	-567	-668	-307	-680	-1341	-306	-400	-374	-397	-889	-187	-975	-1642
Smb protein gene extracted from	X52979_rna1_s_at	1962	1218	2349	971	887	819	2433	2683	1327	2976	1423	1105	4031
GB DEF = mRNA fragment for elon	X03689_s_at	23239	22891	23943	23779	27234	16266	23198	21212	22536	25055	22938	20968	21762
RPS21 Ribosomal protein S21	L04483_s_at	20649	20642	21232	22111	23191	23782	19472	17990	22373	21104	20454	22093	14404
EEF1G Translation elongation fact	X05855_s_at	2583	1991	2527	1465	2812	931	2014	2826	2787	6399	643	1285	7726
PROBABLE PROTEIN DISULFIDE ISC	M13560_s_at	15446	11734	9260	8432	11168	8833	16757	14404	9568	4142	16551	18105	16211
ENO1 Enolase 1, (alpha)	M14328_s_at	2700	6276	9739	4455	3323	6039	7080	10803	7187	13404	14767	7183	10184
PTMA gene extracted from Huma	M14483_rna1_s_a	18443	21771	19363	15326	17346	15270	9946	21908	11204	10465	10737	6475	15861
PTMA Prothymosin alpha	M26708_s_at	13394	14004	13899	17667	16112	11362	13410	12420	13996	13281	13733	13387	12920
VIM Vimentin	Z19554_s_at	15009	16547	18660	9147	17977	4540	15822	14432	17266	16164	9643	3753	16305
Guanine nucleotide-binding prot	M21142_cds2_s_a	11359	8754	7882	9595	10836	5924	7673	11598	10160	10096	4096	6095	12533
T-CELL SURFACE GLYCOPROTEIN C	M23323_s_at	2117	1554	977	2115	3531	1043	1219	937	1599	3421	1371	1321	1771
SAT Spermidine/spermine N1-ace	M24485_s_at	2657	2754	3113	3113	2563	2369	2939	5187	2916	8070	4274	1839	3253
Casein kinase II subunit beta (EC	X57152_rna1_s_at	2464	2099	1885	2197	1346	1294	2476	2932	1864	2237	1733	922	2863

Figure 7.2.3 Selected 50 Genes

7.3 Conversion of data :

Steps for conversion of data , so that it can run on RSES :

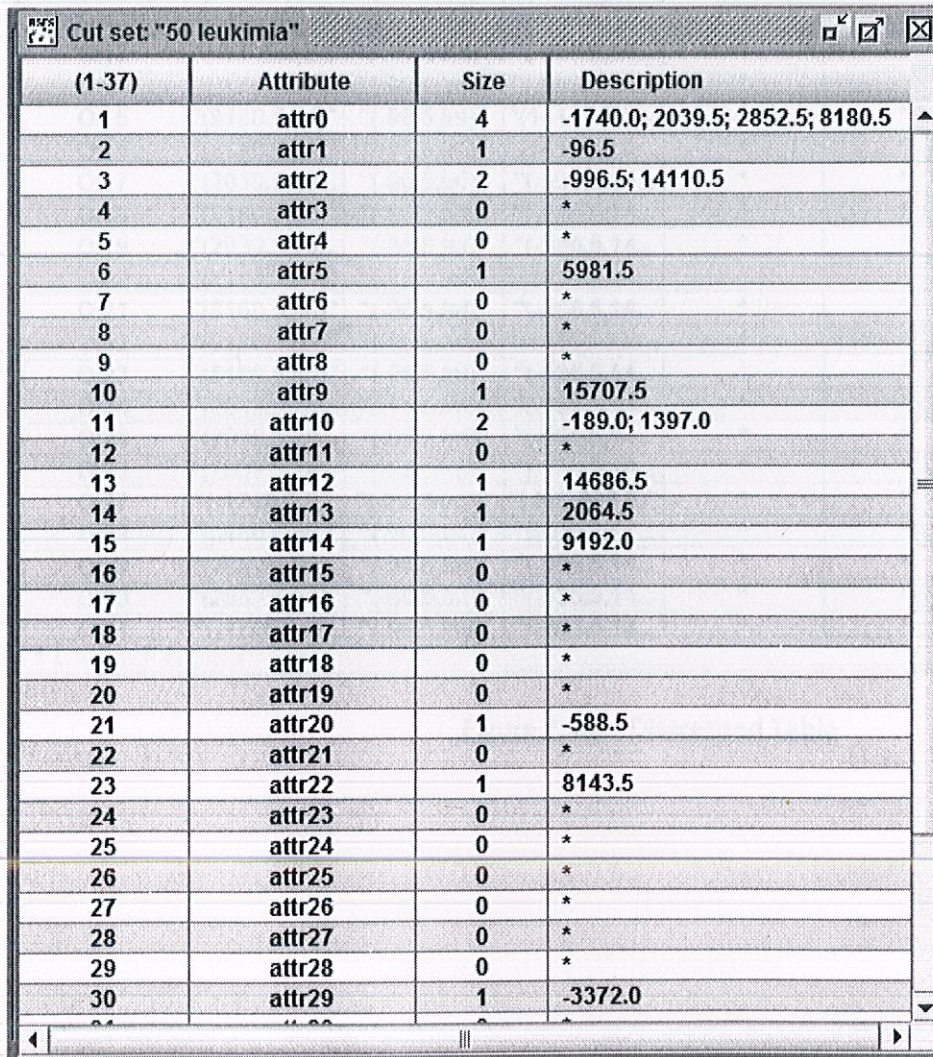
- Data in the excel sheet is saved as Text (Tab delimited).
- Converting the data into .tab file includes changing the data file.
- This file can now be loaded into the table in RSES.

Data Discretization

After the successful clustering of the datasets, we employ the data discretization technique to discretise the continuous data values into meaningful intervals—i.e. nominal values and perform attribute elimination—i.e. attributes that yield only a single discrete value are deemed insignificant and eliminated from the dataset.

7.4 Cuts calculation :

By cuts we understand the definition for decomposition of attribute value sets. In case of numerical attributes being discretized in order to produce a collection of intervals, the cuts are thresholds defining these intervals. In case of symbolic attributes being grouped (quantized), cuts define disjoint subsets of original attribute values.



(1-37)	Attribute	Size	Description
1	attr0	4	-1740.0; 2039.5; 2852.5; 8180.5
2	attr1	1	-96.5
3	attr2	2	-996.5; 14110.5
4	attr3	0	*
5	attr4	0	*
6	attr5	1	5981.5
7	attr6	0	*
8	attr7	0	*
9	attr8	0	*
10	attr9	1	15707.5
11	attr10	2	-189.0; 1397.0
12	attr11	0	*
13	attr12	1	14686.5
14	attr13	1	2064.5
15	attr14	1	9192.0
16	attr15	0	*
17	attr16	0	*
18	attr17	0	*
19	attr18	0	*
20	attr19	0	*
21	attr20	1	-588.5
22	attr21	0	*
23	attr22	1	8143.5
24	attr23	0	*
25	attr24	0	*
26	attr25	0	*
27	attr26	0	*
28	attr27	0	*
29	attr28	0	*
30	attr29	1	-3372.0

Figure 7.4.1 Calculated Cuts

7.5 Descretization :

Discretize/Discretize table option (in data table context menu) makes it possible to discretize (group) attributes in the data table with use of previously calculated cuts. The user sets the set of cuts to be used and the name of object to store resulting discretized table.

Table: "50 leukimia"						
50 / 38	attr0	attr1	attr2	attr3	attr4	attr5
O:1	"(8180.5,Inf)"	"(-96.5,Inf)"	"(14110.5,ln...	*	*	"(5981.5,Inf)"
O:2	"(-1740.0,2...	"(-Inf,-96.5)"	"(-996.5,14...	*	*	"(-Inf,5981....
O:3	"(8180.5,Inf)"	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(5981.5,Inf)"
O:4	"(8180.5,Inf)"	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(5981.5,Inf)"
O:5	"(8180.5,Inf)"	"(-96.5,Inf)"	"(14110.5,ln...	*	*	"(5981.5,Inf)"
O:6	"(8180.5,Inf)"	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(5981.5,Inf)"
O:7	"(2852.5,81...	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(-Inf,5981....
O:8	"(8180.5,Inf)"	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(-Inf,5981....
O:9	"(-1740.0,2...	"(-Inf,-96.5)"	"(-996.5,14...	*	*	"(-Inf,5981....
O:10	"(-1740.0,2...	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(-Inf,5981....
O:11	"(8180.5,Inf)"	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(5981.5,Inf)"
O:12	"(8180.5,Inf)"	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(-Inf,5981....
O:13	"(-1740.0,2...	"(-Inf,-96.5)"	"(-996.5,14...	*	*	"(-Inf,5981....
O:14	"(-1740.0,2...	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(-Inf,5981....
O:15	"(8180.5,Inf)"	"(-96.5,Inf)"	"(14110.5,ln...	*	*	"(5981.5,Inf)"
O:16	"(8180.5,Inf)"	"(-96.5,Inf)"	"(14110.5,ln...	*	*	"(5981.5,Inf)"
O:17	"(2039.5,28...	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(-Inf,5981....
O:18	"(8180.5,Inf)"	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(5981.5,Inf)"
O:19	"(2039.5,28...	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(5981.5,Inf)"
O:20	"(8180.5,Inf)"	"(-96.5,Inf)"	"(14110.5,ln...	*	*	"(5981.5,Inf)"
O:21	"(8180.5,Inf)"	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(5981.5,Inf)"
O:22	"(8180.5,Inf)"	"(-96.5,Inf)"	"(14110.5,ln...	*	*	"(-Inf,5981....
O:23	"(8180.5,Inf)"	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(-Inf,5981....
O:24	"(2039.5,28...	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(-Inf,5981....
O:25	"(2039.5,28...	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(-Inf,5981....
O:26	"(2039.5,28...	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(-Inf,5981....
O:27	"(-1740.0,2...	"(-Inf,-96.5)"	"(-Inf,-996.5)"	*	*	"(-Inf,5981....
O:28	"(8180.5,Inf)"	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(5981.5,Inf)"
O:29	"(2852.5,81...	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(5981.5,Inf)"
O:30	"(2852.5,81...	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(-Inf,5981....
O:31	"(-1740.0,2...	"(-96.5,Inf)"	"(-996.5,14...	*	*	"(-Inf,5981....

Figure 7.5.1 Discretized Table

7.6 Rules Calculation:

Decision rules make it possible to classify objects, i.e. assign the value of decision attribute. Having a collection of rules pointing at different decision we may perform a voting obtaining in this way a simple rule-based decision support system.

(1-50)	Match	Decision rules
1	1	{attr0="(8180.5,Inf)"}&{attr1="-(-96.5,Inf)"}&{attr2="(14110.5,Inf)"}&{attr5="(5981.5,Inf)"}&{attr9="(-Inf,
2	1	{attr0="(-1740.0,2039.5)"}&{attr1="(-Inf,-96.5)"}&{attr2="(-996.5,14110.5)"}&{attr5="(-Inf,5981.5)"}&{
3	1	{attr0="(8180.5,Inf)"}&{attr1="-(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(5981.5,Inf)"}&{attr9=}
4	1	{attr0="(8180.5,Inf)"}&{attr1="(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(5981.5,Inf)"}&{attr9=}
5	1	{attr0="(8180.5,Inf)"}&{attr1="-(-96.5,Inf)"}&{attr2="(14110.5,Inf)"}&{attr5="(5981.5,Inf)"}&{attr9="(15
6	1	{attr0="(8180.5,Inf)"}&{attr1="-(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(5981.5,Inf)"}&{attr9=}
7	1	{attr0="(-2852.5,8180.5)"}&{attr1="(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(-Inf,5981.5)"}&{a
8	1	{attr0="(8180.5,Inf)"}&{attr1="(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(-Inf,5981.5)"}&{attr9=}
9	1	{attr0="(-1740.0,2039.5)"}&{attr1="(-Inf,-96.5)"}&{attr2="(-996.5,14110.5)"}&{attr5="(-Inf,5981.5)"}&{
10	1	{attr0="(-1740.0,2039.5)"}&{attr1="(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(-Inf,5981.5)"}&{a
11	1	{attr0="(8180.5,Inf)"}&{attr1="-(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(5981.5,Inf)"}&{attr9=}
12	1	{attr0="(8180.5,Inf)"}&{attr1="(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(-Inf,5981.5)"}&{attr9=}
13	1	{attr0="(-1740.0,2039.5)"}&{attr1="(-Inf,-96.5)"}&{attr2="(-996.5,14110.5)"}&{attr5="(-Inf,5981.5)"}&{a
14	1	{attr0="(-1740.0,2039.5)"}&{attr1="(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(-Inf,5981.5)"}&{a
15	1	{attr0="(8180.5,Inf)"}&{attr1="(-96.5,Inf)"}&{attr2="(14110.5,Inf)"}&{attr5="(5981.5,Inf)"}&{attr9="(15
16	1	{attr0="(8180.5,Inf)"}&{attr1="-(-96.5,Inf)"}&{attr2="(14110.5,Inf)"}&{attr5="(5981.5,Inf)"}&{attr9="(15
17	1	{attr0="(2039.5,2852.5)"}&{attr1="(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(-Inf,5981.5)"}&{a
18	1	{attr0="(8180.5,Inf)"}&{attr1="(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(5981.5,Inf)"}&{attr9=}
19	1	{attr0="(2039.5,2852.5)"}&{attr1="(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(5981.5,Inf)"}&{at
20	1	{attr0="(8180.5,Inf)"}&{attr1="(-96.5,Inf)"}&{attr2="(14110.5,Inf)"}&{attr5="(5981.5,Inf)"}&{attr9="(-In
21	1	{attr0="(8180.5,Inf)"}&{attr1="(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(5981.5,Inf)"}&{attr9=}
22	1	{attr0="(8180.5,Inf)"}&{attr1="(-96.5,Inf)"}&{attr2="(14110.5,Inf)"}&{attr5="(-Inf,5981.5)"}&{attr9="(1
23	1	{attr0="(8180.5,Inf)"}&{attr1="(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(-Inf,5981.5)"}&{attr9=}
24	1	{attr0="(2039.5,2852.5)"}&{attr1="(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(-Inf,5981.5)"}&{a
25	1	{attr0="(2039.5,2852.5)"}&{attr1="(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(-Inf,5981.5)"}&{a
26	1	{attr0="(2039.5,2852.5)"}&{attr1="(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(-Inf,5981.5)"}&{a
27	1	{attr0="(-1740.0,2039.5)"}&{attr1="(-Inf,-96.5)"}&{attr2="(-Inf,-96.5)"}&{attr5="(-Inf,5981.5)"}&{attr9=}
28	1	{attr0="(8180.5,Inf)"}&{attr1="(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(5981.5,Inf)"}&{attr9=}
29	1	{attr0="(-2852.5,8180.5)"}&{attr1="(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(5981.5,Inf)"}&{af
30	1	{attr0="(-2852.5,8180.5)"}&{attr1="(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(-Inf,5981.5)"}&{a
31	1	{attr0="(-1740.0,2039.5)"}&{attr1="(-96.5,Inf)"}&{attr2="(-996.5,14110.5)"}&{attr5="(-Inf,5981.5)"}&{a

Figure 7.6.1 Calculated Rules

7.7 Reducts Calculation:

Reduct for an information system is a subset of attributes which preserves all discernibility information from the information system, and none of its proper subsets has this ability. These were the reduct sets that were found :

Reduct set: "50 leukemia"				
(1-1)	Size	Pos.Reg.	SC	Reducts
1	12	1	1	{ attr0, attr1, attr2, attr5, attr9, attr10, attr12, attr13, attr14, attr20, attr22, attr29 }

The statistics of these reducts are as follows :

Statistics for reduct set "50 leukemia"

No. reducts: 1

Size of core: 12

Stability coefficient (SC)

Minimal: 1

Maximal: 1

Mean: 1

Length of reducts

Minimal: 12

Maximal: 12

Mean: 12

Show chart

Close

Occurrence of attributes in reducts

Attribute	Count	Percent	Core
attr0	1	8.3	YES
attr1	1	8.3	YES
attr2	1	8.3	YES
attr5	1	8.3	YES
attr9	1	8.3	YES
attr10	1	8.3	YES
attr12	1	8.3	YES

Show chart

As these are all core attributes therefore we separately took these to be considered for classification task :

		ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	AML
AFFX-HSAC07/X00351_s_at	AFFX-HSAC07/X00351	16287	15770	16386	11655	14989	18222	10468	570	8587	22618	18282	21899
XIAA0221 gene	U65533_s_at	-215	-483	-258	-354	-807	-191	-360	-212	-291	-502	-578	-571
Globin, Beta	HG1428-HT1428_s_at	17962	22240	5472	10661	21905	62	9391	25245	25585	23427	1671	15909
RPS3 Ribosomal protein S3	X57351_s_at	16489	10996	12769	16859	25287	1182	20695	26510	4248	7560	7916	14421
LAMR1 Laminin receptor (2)	M14199_s_at	16528	17782	18496	17420	16502	19533	16688	12727	15063	14606	19575	12771
Myosin, Light Chain, Alkali	HG2815-HT2931_s_at	10557	4773	8029	6596	8986	4415	15455	13192	6320	14170	14989	12107
Heterogeneous Nuclear R	HG3076-HT3238_s_at	4094	2621	4087	1244	5765	1351	7284	5034	2641	5095	4843	4530
Unknown protein gene ext	M31520_rna1_s_at	8818	9380	8300	5471	12250	6202	15039	16056	7360	8102	9245	5468
Dna-Binding Protein Hrfx2	HG3327-HT3504_s_at	-1198	-1322	-561	-892	-1557	-640	-834	-598	-932	-1319	-1265	-1115
GB DEF = 52 kD subunit of t	Y07595_at	-385	-67	-203	-280	-568	-289	-417	-166	-290	-570	-425	-658
BETA-2-MICROGLOBULIN P	J00105_s_at	21909	18519	13909	16810	24346	18367	22990	10654	21461	19956	19272	21580
mRNA fragment encoding f	V00599_s_at	16077	11421	10273	5868	19825	15160	13375	5543	7605	21270	21667	17550
SERUM AMYLOID A PROTEI	X51441_s_at	-567	-668	-307	-306	-889	-187	-1642	-208	-545	-607	-394	-525
SmB protein gene extracte	X52979_rna1_s_at	1962	1218	2349	819	2976	1423	4031	694	900	3527	4666	2360
GB DEF = mRNA fragment f	X03689_s_at	23239	22891	23943	16266	25055	22938	21762	164	21744	24547	23594	23716
RPS21 Ribosomal protein S	L04483_s_at	20649	20642	21232	23782	21104	20454	14404	25785	21308	22194	19288	21841
EEFIG Translation elongati	X05855_s_at	2583	1991	2527	931	6399	643	7726	2158	1170	3007	2952	1838
PROBABLE PROTEIN DISULF	M13560_s_at	15446	11734	9260	8033	4142	16551	16211	6256	13625	-662	437	16606
ENO1 Enolase 1, (alpha)	M14328_s_at	2700	6276	9739	6039	13404	14767	10184	2394	1082	6978	12124	16265
PTMA gene extracted from	M14483_rna1_s_at	18443	21771	19363	15270	10465	10737	15861	9913	26892	22160	20381	14909
PTMA Prothymosin alpha	M26708_s_at	13394	14004	13899	11362	13281	13733	12920	14896	12682	15727	12874	14714
VIM Vimentin	Z19554_s_at	15009	16547	18660	4540	16164	9643	16305	18407	12688	18950	20588	25568
Guanine nucleotide-bindin	M21142_cds2_s_at	11359	8754	7882	5924	10096	4096	12533	8800	9227	10402	11291	7937
T-CELL SURFACE GLYCOPRO	M23323_s_at	2117	1554	977	1043	3421	1371	1771	753	1787	6288	9223	2236
SAT Spermidine/spermine	M24485_s_at	2657	2754	3113	2368	8070	4274	3253	795	1404	4739	8371	7022
Casein kinase II subunit be	X57152_rna1_s_at	2464	2099	1885	1294	2237	1733	2863	1445	946	1618	3455	2171

Now by carefully considering the expression values of each gene with respect to ALL and AML from these reducts , we found only those genes that were being classified by their expression level and found these rules :

Unknown protein gene extracted from Human ribosomal protein S24 mRNA	M31520_rna1_s_at	AML	<	5468
GB DEF = 52 kD subunit of transcription factor TFIIF	Y07595_at	AML	<	-658
PROBABLE PROTEIN DISULFIDE ISOMERASE ER-60 PRECURSOR	M13560_s_at	AML	>	16606
ENO1 Enolase 1, (alpha)	M14328_s_at	AML	>	16265
VIM Vimentin	Z19554_s_at	AML	>	25568
Zyxin	X95735_at	AML	>	4871
GB DEF = Polyadenylate binding protein II	Z48501_s_at	AML	>	17637
GB DEF = Neurofilament triplet L protein mRNA, partial cds	U57341_r_at	AML	>	7591
GB DEF = TNNT2 gene exon 11	X98482_r_at	AML	>	17935
SAT Spermidine/spermine N1-acetyltransferase	U21689_at	AML	<	-3381
ACTB Actin, beta	X00351_f_at	AML	>	26861
Major Histocompatibility Complex, Class I, C (Gb:X58536)	HG658-HT658_f_at	AML	>	16619

7.8 Evaluating with the Test set :

Checking only the selected genes from above to check the percentage of classification , We found that :

		ALL	percent correctness
Zyxin	X95735_at	17	80.95238095
GB DEF = 52 kD subunit of transcription factor TFIIH	Y07595_at	5	23.80952381
Unknown protein gene extracted from Human ribosomal protein S24 mRNA	M31520_rna1_s_at	0	0
PROBABLE PROTEIN DISULFIDE ISOMERASE ER-60 PRECURSOR	M13560_s_at	0	0
ENO1 Enolase 1, (alpha)	M14328_s_at	12	57.14285714
VIM Vimentin	Z19554_s_at	0	0
GB DEF = Polyadenylate binding protein II	Z48501_s_at	2	9.523809524
GB DEF = Neurofilament triplet L protein mRNA, partial cds	U57341_r_at	1	4.761904762
GB DEF = TNNT2 gene exon 11	X98482_r_at	3	14.28571429
SAT Spermidine/spermine N1-acetyltransferase	U21689_at	0	0
ACTB Actin, beta	X00351_f_at	0	0
Major Histocompatibility Complex, Class I, C (Gb:X58536)	HG658-HT658_f_at	0	0

As the percentage of classification is highest for Zyxin and the second highest is for ENO1 Enolase 1,(alpha). Therefore these genes are now selected for showing the highest classification. A box plot of X95735 expression levels in the training set is as follows :

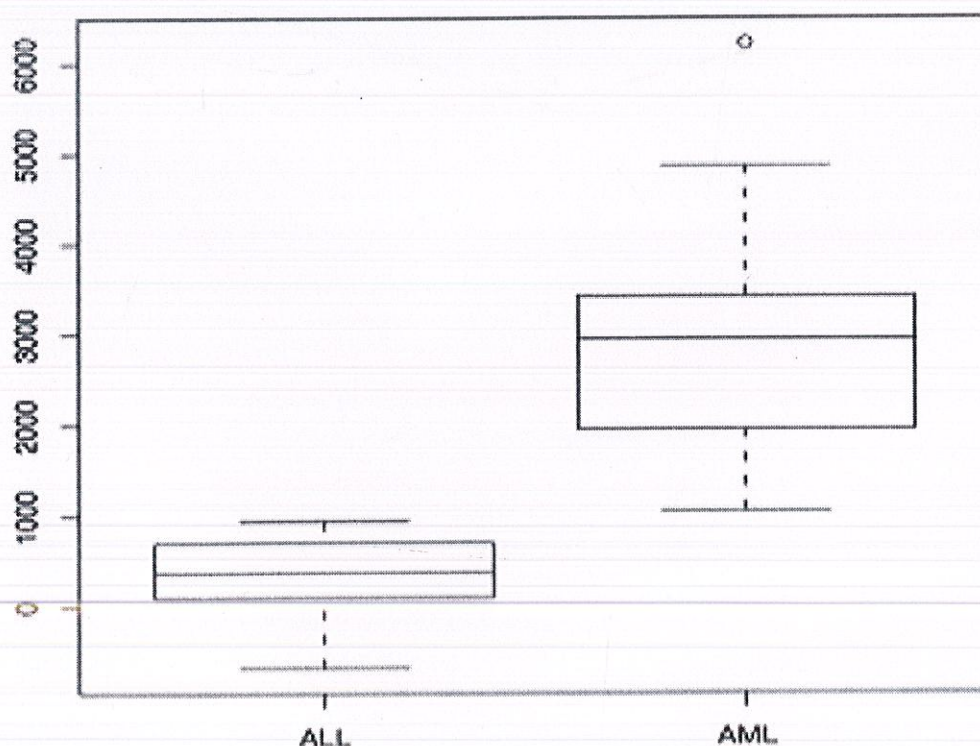


Figure 7.8.1 Box Plot of Zyxin

This figure clearly indicates that the expression levels of X95735 can be used to distinguish ALL from AML in the training set.

Two rules are induced by Rough sets:

If expression level of Zyxin is =>

(A) Higher than 983 than AML

(B) Lower than 983 than ALL

Zyxin is also selected by many other methods. It is reported in "Gene Selection from Microarray Data for Cancer Classification-A Machine Learning Approach", that Zyxin is the only gene identified by J48 pruned tree and the emerging patterns algorithm. Zyxin is repeated selected, and the classification accuracy is 91.2% on test data set. The results obtained by us suggest that the expression level of Zyxin plays an important role in distinguishing two types of acute leukemia. Role of Zyxin in discerning between two types of acute leukemia samples is also verified by biological researchers in Role of Zyxin in "Differential Cell Spreading and Proliferation of Melanoma Cells and Melanocytes."

Chapter-8

Conclusion

Gene expression data set usually has thousands of genes while a few dozens of samples, among a large amount of genes, only a very small fraction of them are informative for classification task. In order to achieve good classification performance, and obtain more useful insight about the biological related issues in cancer classification, gene selection should be well explored to reduce the noise and avoid overfitting of classification algorithm.

The theory of rough sets is a major mathematical tool for managing uncertainty that arises from granularity in the domain of discourse—that is, from the indiscernibility between objects in a set. Rough sets have been applied mainly in mining tasks like classification, clustering and feature selection. A quick search of biological literatures shows that rough sets are still seldom used in bioinformatics.

Systematic and unbiased approach to cancer classification is of great importance to cancer treatment and drug discovery. Previous cancer classification methods are all clinical based and were limited in their diagnostic ability. It has been known that gene expressions contains the keys to the fundamental problems of cancer diagnosis, cancer treatment and drug discovery.

We have performed a successful gene selection method based on rough sets theory. Filter kind of method (clustering by the help of K-mean algorithm) is done first as a preprocessing to select top ranked genes; the minimal reduct of the filtered attribute sets is induced by rough sets. Acute leukemia gene expression dataset is used to test the performance of this novel method; only one gene Zyxin is selected, and high prediction accuracies have been achieved on the test data set. Zyxin is also selected by many other methods, and has been verified by biological researchers to play an important role in distinguish two different types of acute leukemia, AML and ALL.

BIBLIOGRAPHY

1. Z. Pawlak. **Rough Sets: Theoretical Aspects of Reasoning about Data**, Kluwer , McGraw Hill Co. New York 1991
2. Lijun Sun¹, Duoqian Miao² & Hongyun Zhang. **Efficient Gene Selection with Rough Sets from Gene Expression Data**. Rough Sets and Knowledge Technology - RSKT , pp. 164-171, 2008
3. Jan G. Bazan , Hung Son Nguyen , Sinh Hoa Nguyen , Piotr Synak & Jakub Weoblewski **Rough Set Algorithms in classification problem** . Rough set methods and applications
4. **RSES software**. Logic Group , Warsaw University , Poland
5. Mei-Ling Hou, Shu-Lin Wang, Xue-Ling Li,¹ & Ying-Ke Lei. **Neighborhood Rough Set Reduction-Based Gene Selection and Prioritization for Gene Expression Profile Analysis and Molecular Cancer Classification**. Journal of Biomedicine and Biotechnology Volume 2010 (2010), Article ID 726413, 12 pages
6. Syed Sibte Raza Abidi, Kok Meng Hoe & Alwyn Goh. **Analyzing Data Clusters: A Rough Set Approach to Extract Cluster-Defining Symbolic Rules**. Lecture Notes in Computer Science.
7. Lijun Sun, Duoqian Miao & Hongyun Zhang. **Gene Selection with Rough Sets for Cancer Classification** . Fuzzy Systems and Knowledge Discovery , pp. 167-172, 2007.
8. Ying Lu & Jiawei Han. **Cancer Classification Using Gene Expression Data**. Information Systems - IS , vol. 28, no. 4, pp. 243-268.
9. Xiaosheng Wang and Osamu Gotoh. **Cancer Classification Using Single Genes**. Genome Inform. 2009 Oct;23(1):179-88.
10. J. Jeba Emilyn and Dr. K. Ramar. **Rough Set Based Clustering of Gene Expression Data: A Survey**. International Journal of Engineering Science and Technology Vol. 2 (12), 2010, 7160-7164.
11. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield & E. S. Lander. **Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring**. Science , vol. 286, no. 5439, pp. 531-537, 1999.
12. Debahuti Mishra & Dr. Amiya Kumar Rath. **Rough ACO: A Hybridized Model for Feature Selection in Gene Expression Data**. Int. J. of Computer Communication and Technology, Vol. 1, No. 1, 2009.

13. Parvesh Kumar & Siri Krishan Wasan. **Comparative Analysis of k-mean Based Algorithms.** IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.4, April 2010 314.
14. Wai-Ho Au, Keith C. C. Chan, Andrew K. C& Yang Wang. **Attribute Clustering for Grouping, Selection, and Classification of Gene Expression Data.** IEEE.
15. Yu Wanga, Igor V. Tetkoa, Mark A. Hallb, Eibe Frankb, Axel Faciusa, Klaus F.X. Mayera, Hans W. Mewesa,c . **Gene selection from microarray data for cancer classification—a machine learning approach.** Computational Biology and Chemistry 29 (2005) 37–46.
16. Chris Ding & Hanchuan Peng. **Minimum Redundancy Feature Selection from Microarray Gene Expression Data.** J Bioinform Comput Biol. 2005 Apr;3(2):185-205.