Pradipta Maji
Sushmita Paul

Scalable Pattern Recognition Algorithms

Applications in Computational Biology and Bioinformatics

Springer

Scalable Pattern Recognition Algorithms

Pradipta Maji · Sushmita Paul

Scalable Pattern Recognition Algorithms

Applications in Computational Biology and Bioinformatics

Springer

Pradipta Maji
Indian Statistical Institute
Kolkata, West Bengal
India

Sushmita Paul
Indian Statistical Institute
Kolkata, West Bengal
India

ISBN 978-3-319-05629-6 ISBN 978-3-319-05630-2 (eBook)
DOI 10.1007/978-3-319-05630-2
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014933668

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To my daughter

Pradipta Maji

To my parents

Sushmita Paul

Foreword

It is my great pleasure to welcome a new book on "Scalable Pattern Recognition Algorithms: Applications in Computational Biology and Bioinformatics" by Prof. Pradipta Maji and Dr. Sushmita Paul.

This book is unique in its character. Most of the methods presented in it are based on profound research results obtained by the authors. These results are closely related to the main research directions in bioinformatics. The existing conventional/traditional approaches and techniques are also presented, wherever necessary. The effectiveness of algorithms that are proposed by the authors is thoroughly discussed along with both quantitative and qualitative comparisons with other existing methods in this area. These results are derived through experiments on real-life data sets. In general, the presented algorithms display excellent performance. One of the important aspects of the methods proposed by the authors is their ability to scale well with the inflow data. It shall be mentioned that the authors provide in each chapter the directions for future research in the corresponding area.

The main aim of bioinformatics is the development and application of computational methods in pursuit of biological discoveries. Among the hot topics in this field are: sequence alignment and analysis, gene finding, genome annotation, protein structure alignment and prediction, classification of proteins, clustering and dimensionality reduction of gene expression data, protein–protein docking or interactions, and modeling of evolution. From a more general view, the aim is to discover unifying principles of biology using tools of automated knowledge discovery. Hence, knowledge discovery methods that rely on pattern recognition, machine learning, and data mining are widely used for analysis of biological data, in particular for classification, clustering, and feature selection.

The book is structured according to the major phases of a pattern recognition process (clustering, classification, and feature selection) with a balanced mixture of theory, algorithms, and applications. Special emphasis is given to applications in computational biology and bioinformatics.

The reader will find in the book a unified framework describing applications of soft computing, statistical, and machine learning techniques in construction of efficient data models. Soft computing methods allow us to achieve high quality

solutions for many real-life applications. The characteristic features of these methods are tractability, robustness, low-cost solution, and close resemblance with humanlike decision making. They make it possible to use imprecision, uncertainty, approximate reasoning, and partial truth in searching for solutions. The main research directions in soft computing are related to fuzzy sets, neurocomputing, genetic algorithms, probabilistic reasoning, and rough sets. By integration or combination of the different soft computing methods, one may improve the performance of these methods.

The authors of the book present several newly developed methods and algorithms that combine statistical and soft computing approaches, including: (i) neural network tree (NNTree) used for identification of splice-junction and protein coding region in DNA sequences; (ii) a new approach for selecting miRNAs from microarray expression data integrating the merit of rough set-based feature selection algorithm and theory of *B*. 632+ bootstrap error rate; (iii) a robust thresholding technique for segmentation of brain MR images based on the fuzzy thresholding technique; (iv) an efficient method for selecting set of bio-basis strings for the new kernel function, integrating the Fisher ratio and a novel concept of degree of resemblance; (v) a rough set-based feature selection algorithm for selecting sets of effective molecular descriptors from a given quantitative structure activity relationship (QSAR) data set.

Clustering is one of the important analytic tools in bioinformatics. There are several new clustering methods presented in the book. They achieve very good results on various biomedical data sets. That includes, in particular: (i) a method based on Pearson's correlation coefficient that selects initial cluster centers, thus enabling the algorithm to converge to optimal or nearly optimal solution and helping to discover co-expressed gene clusters; (ii) a method based on Dunn's cluster validity index that identifies optimal parameter values during initialization and execution of the clustering algorithm; (iii) a supervised gene clustering algorithm based on the similarity between genes measured with use of the new quantitative measure, whereby redundancy among the attributes is eliminated; (iv) a novel possibilistic biclustering algorithm for finding highly overlapping biclusters having larger volume and mean squared residue lower than a predefined threshold.

The reader will also find several other interesting methods that may be applied in bioinformatics, such as: (i) a computational method for identification of disease-related genes, judiciously integrating the information of gene expression profiles, and the shortest path analysis of protein–protein interaction networks; (ii) a method based on *f*-information measures used in evaluation criteria for gene selection problem.

This book will be useful for graduate students, researchers, and practitioners in computer science, electrical engineering, system science, medical science, bioinformatics, and information technology. In particular, researchers and practitioners in industry and R&D laboratories working in the fields of system design, pattern

recognition, machine learning, computational biology and bioinformatics, data mining, soft computing, computational intelligence, and image analysis may benefit from it.

The authors and editors deserve the highest appreciation for their outstanding work.

Warsaw, Poland, December 2013 Andrzej Skowron

Preface

Recent advancement and wide use of high-throughput technologies for biological research are producing enormous size of biological data distributed worldwide. With the rapid increase in size of biological data banks, understanding the biological data has become critical. Such an understanding could lead us to the elucidation of the secrets of life or ways to prevent certain currently non-curable diseases. Although laboratory experiment is the most effective method for investigating the biological data, it is financially expensive and labor intensive. A deluge of such information coming in the form of genomes, protein sequences, and microarray expression data has led to the absolute need for effective and efficient computational tools to store, analyze, and interpret these multifaceted data.

Bioinformatics is the conceptualizing biology in terms of molecules and applying informatics techniques to understand and organize the information associated with the molecules, on a large scale. It involves the development and advancement of algorithms using techniques including pattern recognition, machine learning, applied mathematics, statistics, informatics, and biology to solve biological problems usually on the molecular level. Major research efforts in this field include sequence alignment and analysis, gene finding, genome annotation, protein structure alignment and prediction, classification of proteins, clustering and dimensionality reduction of microarray expression data, protein–protein docking or interactions, modeling of evolution, and so forth. In other words, bioinformatics can be described as the development and application of computational methods to make biological discoveries. The ultimate attempt of this field is to develop new insights into the science of life as well as creating a global perspective, from which the unifying principles of biology can be derived. As classification, clustering, and feature selection are needed in this field, pattern recognition tools and machine learning techniques have been widely used for analysis of biological data as they provide useful tools for knowledge discovery in this field.

Pattern recognition is the scientific discipline whose goal is the classification of objects into a number of categories or classes. It is the subject of researching object description and classification method. It is also a collection of mathematical, statistical, heuristic, and inductive techniques of the fundamental role in executing the tasks like human beings on computers. In a general setting, the process of pattern recognition is visualized as a sequence of a few steps: data acquisition; data preprocessing; feature selection; and classification or clustering. In the first step,

data are gathered via a set of sensors depending on the environment within which the objects are to be classified. After data acquisition phase, some preprocessing tasks such as noise reduction, filtering, encoding, and enhancement are applied on the collected data for extracting pattern vectors. Afterward, a feature space is constituted to reduce the space dimensionality. However, in a broader perspective this stage significantly influences the entire recognition process. Finally, the classifier is constructed, or in other words, a transformation relationship is established between features and classes.

Pattern recognition, by its nature, admits many approaches, sometimes complementary, sometimes competing, to provide the appropriate solution for a given problem. For any pattern recognition system, one needs to achieve robustness with respect to random noise and failure of components and to obtain output in real time. It is also desirable for the system to be adaptive to the changes in the environment. Moreover, a system can be made artificially intelligent if it is able to emulate some aspects of the human reasoning system. Soft computing and machine learning approaches to pattern recognition are attempts to achieve these goals. Artificial neural network, genetic algorithms, fuzzy sets, and rough sets are used as the tools in these approaches. The challenge is, therefore, to devise powerful pattern recognition methodologies by symbiotically combining these tools for analyzing biological data in more efficient ways. The systems should have the capability of flexible information processing to deal with real-life ambiguous situations and to achieve tractability, robustness, and low-cost solutions.

Various scalable pattern recognition algorithms using soft computing and machine learning approaches, and their real-life applications, including those in computational biology and bioinformatics, have been reported during the last 5–7 years. These are available in different journals, conference proceedings, and edited volumes. This scattered information causes inconvenience to readers, students, and researchers. The current volume is aimed at providing a treatise in a unified framework describing how soft computing and machine learning techniques can be judiciously formulated and used in building efficient pattern recognition models. Based on the existing as well as new results, the book is structured according to the major phases of a pattern recognition system (classification, feature selection, and clustering) with a balanced mixture of theory, algorithm, and applications. Special emphasis is given to applications in computational biology and bioinformatics.

The book consists of 11 chapters. Chapter 1 provides an introduction to pattern recognition and bioinformatics, along with different research issues and challenges related to high-dimensional real-life biological data sets. The significance of pattern recognition and machine learning techniques in computational biology and bioinformatics is also presented in Chap. 1. Chapter 2 presents the design of a hybrid learning model, termed as neural network tree (NNTree), for identification of splice-junction and protein coding region in DNA sequences. It incorporates the advantages of both decision tree and neural network. An NNTree is a decision tree, where each non-terminal node contains a neural network. The versatility of this method is illustrated through its application in splice-junction and gene

identification problems. Extensive experimental results establish that the NNTree produces more accurate classifier than that previously obtained for a range of different sequence lengths, thereby indicating a cost-effective alternative in splice-junction and protein coding region identification problem.

The prediction of protein functional sites is an important issue in protein function studies and drug design. In order to apply the powerful kernel-based pattern recognition algorithms such as support vector machine to predict functional sites in proteins, amino acids need encoding prior to input. In this regard, a new string kernel function, termed as the modified bio-basis function, is presented in Chap. 3. It maps a nonnumerical sequence space to a numerical feature space using a bio-basis string as its support. The concept of zone of influence of bio-basis string is introduced in the new kernel function to take into account the influence of each bio-basis string in nonnumerical sequence space. An efficient method is described to select a set of bio-basis strings for the new kernel function, integrating the Fisher ratio and the concept of degree of resemblance. The integration enables the method to select a reduced set of relevant and nonredundant bio-basis strings. Some quantitative indices are described for evaluating the quality of selected bio-basis strings. The effectiveness of the new string kernel function and bio-basis string selection method, along with a comparison with existing bio-basis function and related bio-basis string selection methods, is demonstrated on different protein data sets using the new quantitative indices and support vector machine.

Quantitative structure activity relationship (QSAR) is one of the important disciplines of computer-aided drug design that deals with the predictive modeling of properties of a molecule. In general, each QSAR data set is small in size with a large number of features or descriptors. Among the large amount of descriptors present in the QSAR data set, only a small fraction of them is effective for performing the predictive modeling task. Chapter 4 presents a rough set-based feature selection algorithm to select a set of effective molecular descriptors from a given QSAR data set. The new algorithm selects the set of molecular descriptors by maximizing both relevance and significance of the descriptors. The performance of the new algorithm is studied using the R^2 statistic of support vector regression method. The effectiveness of the new algorithm, along with a comparison with existing algorithms, is demonstrated on several QSAR data sets.

Microarray technology is one of the important biotechnological means that allows to record the expression levels of thousands of genes simultaneously within a number of different samples. An important application of microarray gene expression data in functional genomics is to classify samples according to their gene expression profiles. Among the large amount of genes present in microarray gene expression data, only a small fraction of them is effective for performing a certain diagnostic test. In this regard, mutual information has been shown to be successful for selecting a set of relevant and nonredundant genes from microarray data. However, information theory offers many more measures such as the f-information measures that may be suitable for selection of genes from microarray gene expression data.

Chapter 5 presents different *f*-information measures as the evaluation criteria for gene selection problem. The performance of different *f*-information measures is compared with that of mutual information based on the predictive accuracy of naive Bayes classifier, *k*-nearest neighbor rule, and support vector machine. An important finding is that some *f*-information measures are shown to be effective for selecting relevant and nonredundant genes from microarray data. The effectiveness of different *f*-information measures, along with a comparison with mutual information, is demonstrated on several cancer data sets.

One of the most important and challenging problems in functional genomics is how to select the disease genes. In Chap. 6, a computational method is reported to identify disease genes, judiciously integrating the information of gene expression profiles and shortest path analysis of protein–protein interaction networks. While the gene expression profiles have been used to select differentially expressed genes as disease genes using mutual information-based maximum relevance-maximum significance framework, the functional protein association network has been used to study the mechanism of diseases. Extensive experimental study on colorectal cancer establishes the fact that the genes identified by the integrated method have more colorectal cancer genes than the genes identified from the gene expression profiles alone. All these results indicate that the integrated method is quite promising and may become a useful tool for identifying disease genes.

The microRNAs or miRNAs regulate expression of a gene or protein. It has been observed that they play an important role in various cellular processes and thus help in carrying out normal functioning of a cell. However, dysregulation of miRNAs is found to be a major cause of a disease. Various studies have also shown the role of miRNAs in cancer and utility of miRNAs for the diagnosis of cancer. In this regard, Chap. 7 presents a new approach for selecting miRNAs from microarray expression data. It integrates the merit of rough set-based feature selection algorithm reported in Chap. 4 and theory of B. 632+ bootstrap error rate. The effectiveness of the new approach, along with a comparison with other algorithms, is demonstrated on several miRNA data sets.

Clustering is one of the important analyses in functional genomics that discovers groups of co-expressed genes from microarray data. In Chap. 8, different partitive clustering algorithms such as hard *c*-means, fuzzy *c*-means, rough-fuzzy *c*-means, and self-organizing maps are presented to discover co-expressed gene clusters. One of the major issues of the partitive clustering-based microarray data analysis is how to select initial prototypes of different clusters. To overcome this limitation, a method is reported based on Pearson's correlation coefficient to select initial cluster centers. It enables the algorithm to converge to an optimum or near optimum solutions and helps to discover co-expressed gene clusters. In addition, a method is described to identify optimum values of different parameters of the initialization method and the clustering algorithm. The effectiveness of different algorithms is demonstrated on several yeast gene expression time-series data sets using different cluster validity indices and gene ontology-based analysis.

In functional genomics, an important application of microarray data is to classify samples according to their gene expression profiles such as to classify

cancer versus normal samples or to classify different types or subtypes of cancer. Hence, one of the major tasks with the gene expression data is to find groups of co-regulated genes whose collective expression is strongly associated with the sample categories or response variables. In this regard, a supervised gene clustering algorithm is presented in Chap. 9 to find groups of genes. It directly incorporates the information about sample categories into the gene clustering process. A new quantitative measure, based on mutual information, is reported that incorporates the information about sample categories to measure the similarity between attributes. The supervised gene clustering algorithm is based on measuring the similarity between genes using the new quantitative measure. The performance of the new algorithm is compared with that of existing supervised and unsupervised gene clustering and gene selection algorithms based on the class separability index and the predictive accuracy of naive Bayes classifier, *k*-nearest neighbor rule, and support vector machine on several cancer and arthritis microarray data sets. The biological significance of the generated clusters is interpreted using the gene ontology.

The biclustering method is another important tool for analyzing gene expression data. It focuses on finding a subset of genes and a subset of experimental conditions that together exhibit coherent behavior. However, most of the existing biclustering algorithms find exclusive biclusters, which is inappropriate in the context of biology. Since biological processes are not independent of each other, many genes may participate in multiple different processes. Hence, nonexclusive biclustering algorithms are required for finding overlapping biclusters. In Chap. 10, a novel possibilistic biclustering algorithm is presented to find highly overlapping biclusters of larger volume with mean squared residue lower than a predefined threshold. It judiciously incorporates the concept of possibilistic clustering algorithm into biclustering framework. The integration enables efficient selection of highly overlapping coherent biclusters with mean squared residue lower than a given threshold. The detailed formulation of the new possibilistic biclustering algorithm, along with a mathematical analysis on the convergence property, is presented. Some quantitative indices are reported for evaluating the quality of generated biclusters. The effectiveness of the algorithm, along with a comparison with other algorithms, is demonstrated on yeast gene expression data set.

Finally, Chap. 11 reports a robust thresholding technique for segmentation of brain MR images. It is based on the fuzzy thresholding techniques. Its aim is to threshold the gray level histogram of brain MR images by splitting the image histogram into multiple crisp subsets. The histogram of the given image is thresholded according to the similarity between gray levels. The similarity is assessed through a second-order fuzzy measure such as fuzzy correlation, fuzzy entropy, and index of fuzziness. To calculate the second-order fuzzy measure, a weighted co-occurrence matrix is presented, which extracts the local information more accurately. Two quantitative indices are reported to determine the multiple thresholds of the given histogram. The effectiveness of the algorithm, along with a comparison with standard thresholding techniques, is demonstrated on a set of brain MR images.

The relevant existing conventional/traditional approaches or techniques are also included wherever necessary. Directions for future research in the concerned topic are provided in each chapter. Most of the materials presented in the book are from our published works. For the convenience of readers, a comprehensive bibliography on the subject is also appended in each chapter. It might have happened that some works in the related areas have been omitted due to oversight or ignorance.

The book, which is unique in its character, will be useful to graduate students and researchers in computer science, electrical engineering, system science, medical science, bioinformatics, and information technology both as a textbook and as a reference book for some parts of the curriculum. The researchers and practitioners in industry and R&D laboratories working in the fields of system design, pattern recognition, machine learning, computational biology and bioinformatics, data mining, soft computing, computational intelligence, and image analysis will also be benefited.

Finally, the authors take this opportunity to thank Mr. Wayne Wheeler and Mr. Simon Rees of Springer-Verlag, London, for their initiative and encouragement. The authors also gratefully acknowledge the support provided by Dr. Chandra Das of Netaji Subhash Engineering College, Kolkata, India and the members of Biomedical Imaging and Bioinformatics Lab, Indian Statistical Institute, Kolkata, India for preparation of a few chapters of the manuscript. The book has been written when one of the authors, Dr. S. Paul, held a CSIR Fellowship of the Government of India. This work is partially supported by the Indian National Science Academy, New Delhi (grant no. SP/YSP/68/2012).

Kolkata, India, January 2014

Pradipta Maji
Sushmita Paul

Contents

Part II Feature Selection

Chapter 1
Introduction to Pattern Recognition and Bioinformatics

1.1 Introduction

With the gaining of knowledge in different branches of biology such as molecular biology, structural biology, and biochemistry, and the advancement of technologies lead to the generation of biological data at a phenomenal rate [286]. The enormous quantity and variety of information are being produced from the data of the myriad of projects that study gene expression, determine the protein structures encoded by the genes, and detail how these products interact with one another. This deluge of biological information has, in turn, led to an absolute need for computerized databases to store, organize, and index the data, and for specialized tools to view and analyze the data. Hence, computers have become indispensable to biological research. Such an approach is ideal due to the ease with which computers can handle large quantities of data and probe the complex dynamics observed in nature.

Bioinformatics is a multidisciplinary research area that conceptualizes biology in terms of molecules and applies information techniques to understand and organize the information associated with these molecules on a large scale. It involves the development and advancement of algorithms using techniques including pattern recognition, machine learning, applied mathematics, statistics, informatics, and biology to analyze the complete collection of DNA (the genome), RNA (the transcriptome), and protein (the proteome) of an organism [275]. Major research efforts in this field include sequence alignment and analysis, gene finding, genome annotation, protein structure alignment and prediction, classification of proteins, clustering and dimensionality reduction of microarray expression data, protein–protein docking or interactions, modeling of evolution, and so forth. In other words, bioinformatics can be described as the development and application of computational methods to make biological discoveries. The ultimate attempt of this field is to develop new insights into the science of life as well as creating a global perspective, from which the unifying principles of biology can be derived [20, 22, 209, 302, 377, 391].

Pattern recognition is the scientific discipline whose goal is the classification of objects into a number of categories or classes. It is the subject of researching

P. Maji and S. Paul, *Scalable Pattern Recognition Algorithms*,
DOI: 10.1007/978-3-319-05630-2_1,

object description and classification method. It is also a collection of mathematical, statistical, heuristic, and inductive techniques of fundamental role in executing the tasks like human being on computers [209, 260, 263]. As classification, clustering, and feature selection are needed in bioinformatics, pattern recognition and machine learning techniques have been widely used for analysis of biological data as they provide useful tools for knowledge discovery in this field. The massive biological databases are generally characterized by the numeric as well as textual, symbolic, and pictorial data. They may contain redundancy, errors, and imprecision. The pattern recognition is aimed at discovering natural structures within such massive and often heterogeneous biological data. It is visualized as being capable of knowledge discovery using generalizations and magnifications of existing and new algorithms. Therefore, pattern recognition plays a significant role in bioinformatics [20, 22, 209, 302, 377, 391]. It deals with the process of identifying valid, novel, potentially useful, and ultimately understandable patterns in voluminous, possibly heterogeneous biological data sets.

One of the main problems in biological data analysis is uncertainty. Some of the sources of this uncertainty include imprecision in computations and vagueness in class definition. Pattern recognition, by its nature, admits many approaches, sometimes complementary, sometimes competing, to provide the appropriate solution of a given problem. An efficient pattern recognition system for bioinformatics tasks should possess several characteristics such as online adaptation to cope with the changes in the environment, handling nonlinear class separability to tackle real-life problems, handling of overlapping classes or clusters for discriminating almost similar but different objects, real-time processing for making a decision in a reasonable time, generation of soft and hard decisions to make the system flexible, verification and validation mechanisms for evaluating its performance, and minimizing the number of parameters in the system that have to be tuned for reducing the cost and complexity. The property to emulate some aspects of the human processing system can be helpful for making a system artificially intelligent.

Soft computing and machine learning approaches to pattern recognition are attempts to achieve these goals. Artificial neural network, genetic algorithms, information theory, fuzzy sets, and rough sets are used as the tools in these approaches. The challenge is, therefore, to devise powerful pattern recognition methodologies by symbiotically combining these tools for analyzing biological data in more efficient ways. The systems should have the capability of flexible information processing to deal with real-life ambiguous situations and to achieve tractability, robustness, and low-cost solutions. Various scalable pattern recognition algorithms using soft computing and machine learning approaches have been developed to successfully address different problems of computational biology and bioinformatics [33, 56, 68, 83, 89, 98, 107, 131, 198, 210, 211, 318, 324, 350, 357, 363–365, 375, 376, 380].

The objective of this book is to provide some results of investigations, both theoretical and experimental, addressing the relevance of information theory, artificial neural networks, fuzzy sets, and rough sets to bioinformatics with real-life applications. Various methodologies are presented based on information theoretic measures, artificial neural networks, fuzzy sets, and rough sets for classification, feature selection, and

clustering. The emphasis of these methodologies is given on (a) handling biological data sets which are large, both in size and dimension, and involve classes that are overlapping, intractable, and/or having nonlinear boundaries, (b) demonstrating the significance of pattern recognition and machine learning for dealing with the biological knowledge discovery aspect, and (c) demonstrating their success in certain tasks of bioinformatics and medical imaging as examples. Before describing the scope of the book, a brief overview of molecular biology and pattern recognition is provided.

The structure of the rest of this chapter is as follows: Section 1.2 briefly presents a description of the basic concept of molecular biology. In Sect. 1.3, several bioinformatics problems are reported, which are important to retrieve useful biological information from large data sets using pattern recognition and machine learning techniques. In Sect. 1.4, the pattern recognition aspect is elaborated, discussing its components, tasks involved, and approaches, along with the role of soft computing in bioinformatics and computational biology. Finally, Sect. 1.5 discusses the scope and organization of the book.

1.2 Basics of Molecular Biology

The molecular biology deals with the formation, structure, and function of macromolecules essential to life, such as carbohydrates, nucleic acids, and proteins, including their roles in cell replication and the transmission of genetic information [190]. This field overlaps with other areas of biology and chemistry, particularly genetics and biochemistry. This section presents the basic concepts of nucleic acids and proteins.

1.2.1 Nucleic Acids

The weakly acidic substance present inside a nuclei is known as nucleic acids. They are large biological molecules essential for all known forms of life. They include deoxyribonucleic acid (DNA) and ribonucleic acid (RNA) [190].

1.2.1.1 DNA

It contains the instructions needed by the cell to carry out its functions [190]. DNA consists of two long interwoven strands that form the famous double helix. Each strand is built from a small set of constituent molecules called nucleotides. The first two parts of the nucleotides are used to form the ribbon-like backbone of the DNA strand, and are identical in all nucleotides. These two parts are a phosphate group and a sugar called deoxyribose. The third part of the nucleotide is the base. There are four different bases, which define the four different nucleotides, namely, thymine

(**T**), cytosine (**C**), adenine (**A**), and guanine (**G**). The base pair complementarity makes a DNA molecule double stranded. If specific bases of one strand are aligned with specific bases on the other strand, the aligned bases can hybridize via hydrogen bonds, weak attractive forces between hydrogen and either nitrogen or oxygen. The specific complementary pairs are **A** with **T** and **G** with **C**. Two hydrogen bonds occur between **A** and **T**, whereas three bonds are formed between **C** and **G**. This makes **C**–**G** bonds stronger than **A**–**T** bonds.

DNA is the genetic material, used in development and functioning of all known living organisms and many viruses. It contains informations that are required to construct other important components of a cell like protein and RNA molecules. This biological information of DNA is decoded with the help of ribosomes, which links amino acids in an order specified by messenger RNA (mRNA), using transfer RNA molecules to carry amino acids and to read the mRNA three nucleotides at a time. The genetic code is highly similar among all organisms, and can be expressed in a simple table with 64 entries. These 64 codons code for 20 different amino acids. The code defines how sequences of these nucleotide triplets, called codons, specify which amino acid will be added next during protein synthesis. Amino acids play central roles both as building blocks of proteins and as intermediates in metabolism. The DNA sequences that code for protein are known as genes, other part of DNA is known as junk DNA. Much of this DNA has no known biological function. However, many types of it do have known biological functions, including the transcriptional and translational regulation of protein coding sequences. A brief description of important components and processes of DNA is as follows [190]:

- *Gene* is a molecular unit of heredity of a living organism. Living beings depend on genes, as they specify all proteins and functional RNA chains. Genes hold the information to build and maintain an organism's cells and pass genetic traits to offspring. All organisms have many genes corresponding to various 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, increased risk for specific diseases, or the thousands of basic biochemical processes that comprise life.
- *Gene expression* is the process by which information from a gene is used in the synthesis of a functional gene product. These products are often proteins, but in nonprotein coding genes such as ribosomal RNA genes or transfer RNA genes, the product is a functional RNA. The process of gene expression is used by all known life—eukaryotes (including multicellular organisms), prokaryotes (bacteria and archaea), and viruses—to generate the macromolecular machinery for life.
- *Transcription* is the process of making an RNA copy of a gene sequence. In a eukaryotic cell, this copy, called mRNA molecule, leaves the cell nucleus and enters the cytoplasm, where it directs the synthesis of the protein, which it encodes. However, in a prokaryotic cell there is no nucleus, so the transcription as well as translation take place in cytoplasm.
- *Translation* is the process of translating the sequence of a mRNA molecule to a sequence of amino acids during protein synthesis. The genetic code describes the relationship between the sequence of base pairs in a gene and the corresponding

amino acid sequence that it encodes. In the cell cytoplasm, the ribosome reads the sequence of the mRNA in groups of three bases to assemble the protein.

1.2.1.2 RNA

The mRNA and other types of RNAs are single-stranded nucleic acids made up of ribose sugar, phosphate group, and nucleobases (**G**, **A**, uracil (**U**), **C**). The genetic information stored in DNA is transferred into RNA through transcription by DNA polymerase, and the information is decoded when RNA is translated into proteins. The proteins largely constitute the machinery that makes life live. They carry out all structural, catalytic, and regulatory functions. Hence, RNAs mostly play the passive role of a messenger. RNAs can be divided into two classes, namely, coding RNA and noncoding RNA.

The RNAs that code for proteins are known as coding RNA. The transcribed coding RNAs, that is, mRNAs are further translated into proteins. The mRNA serves as a template for protein synthesis. It is transcribed from a gene and then translated by ribosomes in order to manufacture a protein. Hence, it is known as coding RNA. The sequence of a strand of mRNA is based on the sequence of a complementary strand of DNA. The RNAs those do not translated into proteins are known as noncoding RNAs. The noncoding RNAs have been found to carry out very diverse functions, from mRNA splicing and RNA modification to translational regulation. MicroRNA (miRNA) is one type of noncoding RNAs. The miRNAs are small noncoding RNAs of length around 22 nucleotides, present in animal and plant cell. They regulate the expression of mRNAs posttranscriptionally, resulting in translational repression and gene silencing. Hence, miRNAs are related to diverse cellular processes and regarded as important components of the gene regulatory network [275].

1.2.2 Proteins

Proteins are organic compounds made of amino acids arranged in a linear chain and folded into a globular or fibrous form [185]. The amino acids in a polymer are joined together by the peptide bonds between the carboxyl and amino groups of adjacent amino acid residues. The sequence of amino acids in a protein is defined by the sequence of a gene, which is encoded in the genetic code. Amino acids can be divided into two groups, namely, essential amino acids and nonessential amino acids. The liver, and to a much lesser extent the kidneys, can convert amino acids used by cells in protein biosynthesis into glucose by a process known as gluconeogenesis. The essential amino acids, which must be obtained from external sources such as food, are leucine, isoleucine, valine, lysine, threonine, tryptophan, methionine, phenylalanine, and histidine. On the other hand, nonessential amino acids are synthesized in our body from other amino acids. The nonessential amino acids are arginine, alanine, asparagine, aspartic acid, cysteine, glutamine, glutamic acid, glycine, proline,

serine, and tyrosine. In the form of skin, hair, callus, cartilage, muscles, tendons, and ligaments, proteins hold together, protect, and provide structure to the body of a multicelled organism. In the form of enzymes, hormones, antibodies, and globulins, they catalyze, regulate, and protect the body chemistry. In the form of hemoglobin, myoglobin, and various lipoproteins, they effect the transport of oxygen and other substances within an organism.

1.3 Bioinformatics Tasks for Biological Data

This section presents the major biological problems and associated tasks involved in computational biology and bioinformatics.

1.3.1 Alignment and Comparison of DNA, RNA, and Protein Sequences

An alignment is a mutual placement of two or more sequences which exhibit where the sequences are similar, and where they differ. These include alignment and prediction of DNA, RNA, protein sequences, and fragment assembly of DNA. An optimal alignment is the one that exhibits the most correspondences and the fewest differences. There are mainly two types of alignment methods, namely, global alignment and local alignment. Global alignment [239] maximizes the number of matches between the sequences along the entire length of the sequence, while local alignment [325] gives a highest scoring to local match between two sequences. Global alignment includes all the characters in both sequences from one end to the other, and is excellent for sequences that are known to be very similar. If the sequences being compared are not similar over their entire lengths, but have short stretches within them that have high levels of similarity, a global alignment may miss the alignment of these important regions, and local alignment is then used to find these internal regions of high similarity.

Pairwise comparison and alignment of protein or nucleic acid sequences is the foundation upon which most other bioinformatics tools are built. Dynamic programming is an algorithm that allows for efficient and complete comparison of two or more biological sequences, and the technique is known as the Smith–Waterman algorithm [325]. It refers to a programmatic technique or algorithm which, when implemented correctly, effectively makes all possible pairwise comparisons between the characters (nucleotide or amino acid residues) in two biological sequences. Spaces may need to be inserted within the sequences for alignment. Consecutive space is defined as a gap. The final result is a mathematical, but not necessarily biological, optimal alignment of the two sequences. A similarity score is also generated to describe how

similar the two sequences are, given the specific parameters used. A few of the many popular alignment techniques are BLAST [7], FASTA [272], and PSI-BLAST [8].

A multiple alignment [242] arranges a set of sequences in a manner that positions homologous sequences in a common column. There are different conventions regarding the scoring of a multiple alignment. In one approach, the scores of all the induced pairwise alignments contained in a multiple alignment are simply added. For a linear gap penalty, this amounts to scoring each column of the alignment by the sum of pair scores in this column [308]. Although it would be biologically meaningful, the distinctions between global, local, and other forms of alignment are rarely made in a multiple alignment. A full set of optimal pairwise alignments among a given set of sequences will generally overdetermine the multiple alignment. If one wishes to assemble a multiple alignment from pairwise alignments, one has to avoid closing loops, that is, one can put together pairwise alignments as long as no new pairwise alignment is included to a set of sequences which is already part of the multiple alignment.

1.3.2 Identification of Genes and Functional Sites from DNA Sequences

Gene finding is concerned with identifying stretches of sequence, usually genomic DNA, that are biologically functional. This especially includes identification of protein coding genes, but may also include identification of other functional elements such as noncoding RNA genes and regulatory regions. Since in human body the protein coding regions account for only a few percent of the total genomic sequence, identifying protein coding genes within large regions of uncharacterized DNA is a difficult task. In bacterial DNA, each protein is encoded by a contiguous fragment called an open reading frame, beginning with a start codon and ending with a stop codon. In eukaryotes, especially in vertebrates, the coding region is split into several fragments called exons, and the intervening fragments are called introns. So, finding eukaryotic protein coding genes in uncharacterized DNA sequences is essentially predicting exon–intron structures. Different works related to identification of protein coding genes are discussed in [99, 101, 102, 348].

Another important problem in bioinformatics is the identification of several functional sites in genomic DNA such as splice sites or junctions, start and stop codons, branch points, promoters and terminators of transcription, polyadenylation sites, topoisomerase II binding sites, topoisomerase I cleavage sites, and various transcription factor-binding sites. Such local sites are called signals, and the methods for detecting them are called signal sensors. Genomic DNA signals can be contrasted with extended and variable length regions such as exons and introns, which are recognized by different methods called content sensors. Identification of splice sites, introns, exons, start and stop codons, and branch points constitutes the major

subtask in gene prediction and is of key importance in determining the exact structure of genes in genomic sequences.

In order to study gene regulation and have a better interpretation of microarray expression data, promoter prediction, and transcription factor-binding site's (TFBS) discovery have become important. A cell mechanism recognizes the beginning of a gene or gene cluster with the help of a promoter and is necessary for the initiation of transcription. The promoter is a region before each gene in the DNA that serves as an indication to the cellular mechanism that a gene is ahead. There exist a number of approaches that find differences between sets of known promoter and nonpromoter sequences [171, 189]. Due to the lack of robust protein coding signatures, current promoter predictions are much less reliable than protein coding region predictions. Once regulatory regions, such as promoters, are obtained, finding the TFBS motifs within these regions may proceed either by enumeration or by alignment to find the enriched motifs. Recognition of regulatory sites in DNA fragments has become particularly popular because of the increasing number of completely sequenced genomes and mass application of DNA chips. Experimental analyses have identified fewer than 10 % of the potential promoter regions, assuming that there are at least 30,000 promoters in the human genome, one for each gene.

1.3.3 Prediction of Protein Functional Sites

The prediction of functional sites in proteins is another important problem in bioinformatics. It is an important issue in protein function studies and hence, drug design. The problem of functional sites prediction deals with the subsequences; each subsequence is obtained through moving a fixed length sliding window residue by residue. The residues within a scan form a subsequence. If there is a match between a subsequence and a consensus pattern of a specific function, a functional site is then identified within the subsequence or the subsequence is labeled as functional, otherwise nonfunctional. To analyze protein sequences, BLAST [7], FASTA [272], PSI-BLAST [8], suffix-tree based algorithms [4], regular expression matching representations [337], and finite state machines [304, 305] are a few of the many pattern recognition algorithms that use characters or strings as their primitive type.

However, it has been found that the relation between functional sites and consensus patterns may not be always simple and the development and the use of more complicated and hence, more powerful pattern recognition algorithms is a necessity. The artificial neural networks trained with backpropagation [55, 236, 280], Kohonen's self-organizing map [13], feedforward and recurrent neural networks [19, 20], biobasis function neural networks [38, 338, 376, 378–380], and support vector machine [56, 226, 375] have been widely used to predict different functional sites in proteins such as protease cleavage sites of HIV (human immunodeficiency virus) and Hepatitis C Virus, linkage sites of glycoprotein, enzyme active sites, post-translational phosphorylation sites, immunological domains, Trypsin cleavage sites, protein–protein interaction sites, and so forth.

1.3.4 DNA and RNA Structure Prediction

DNA structure plays an important role in a variety of biological processes. Different dinucleotide and trinucleotide scales have been described to capture various aspects of DNA structure including base stacking energy, propeller twist angle, protein deformability, bendability, and position preference [19]. Three dimensional DNA structure and its organization into chromatin fibers are essential for its functions, and are applied in protein binding sites, gene regulation, and triplet repeat expansion diseases.

An RNA molecule is considered as a string of n characters $R = r_1 r_2 \cdots r_n$ such that $r_i \in \{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{U}\}$. Typically, n is in the hundreds, but could also be in thousands. The secondary structure of the RNA molecule is a collection S of a set of stems and each stem consisting of a set of consecutive base pairs $(r_i r_j)$ (for example, **GU**, **GC**, **AU**). Here, $1 \leq i \leq j \leq n$ and (r_i and r_j) are connected through hydrogen bonds. If $(r_i, r_j) \in S$, in principle, we should require that r_i be a complement to r_j and that $j - i > t$, for a certain threshold t as it is known that an RNA molecule does not fold too sharply on itself.

Attempts to automatically predict the RNA secondary structure can be divided in essentially two general approaches. The first involves the overall free energy minimization by adding contributions from each base pair, bulged base, loop, and other elements [1]. The second type of approach [360] is more empirical and it involves searching for the combination of nonexclusive helices with a maximum number of base pairings, satisfying the condition of a tree-like structure for the biomolecule. Within the latter, methods using dynamic programming are the most common [360, 395]. The methods for simulating the folding pathway of an RNA molecule [312, 313, 366] and locating significant intermediate states are important for the prediction of RNA structure [29, 127, 311] and its associated function.

1.3.5 Protein Structure Prediction and Classification

Identical protein sequences result in identical 3D structures. So, it follows that similar sequences may result in similar structures, and this is usually the case. However, identical 3D structures do not necessarily indicate identical sequences as there is a distinction between homology and similarity. There are a few examples of proteins in the databases that have nearly identical 3D structures, and are therefore homologous, but do not exhibit significant or detectable sequence similarity. Pairwise comparisons do not readily show positions that are conserved among a whole set of sequences and tend to miss subtle similarities that become visible when observed simultaneously among many sequences. Hence, one wants to simultaneously compare several sequences. Structural genomics is the prediction of the 3D structure of a protein from the primary amino acid sequence [21, 60, 70, 73, 112, 128, 150, 166, 175, 219,

220, 245, 268, 280, 287, 294, 295, 297, 329]. This is one of the most challenging tasks in bioinformatics as a protein's function is a consequence of its structure.

There are five levels of protein structure. While the primary structure is the sequence of amino acids that compose the protein, the secondary structure of a protein is the spatial arrangement of the atoms constituting the main protein backbone. The supersecondary structure or motif is the local folding pattern built up from particular secondary structures. On the other hand, tertiary structure is formed by packing secondary structural elements linked by loops and turns into one or several compact globular units called domains, that is, the folding of the entire protein chain. A final protein may contain several protein subunits arranged in a quaternary structure.

Protein sequences almost always fold into the same structure in the same environment. Hydrophobic interaction, hydrogen bonding, electrostatic, and other van der Waals type interactions also contribute to determine the structure of the protein. Many efforts are underway to predict the structure of a protein, given its primary sequence. A typical computation of protein folding would require computing all the spatial coordinates of atoms in a protein molecule, starting with an initial configuration and working up to a final minimum-energy folding configuration [31, 74, 174, 176, 273, 284, 303, 349]. Sequence similarity methods can predict the secondary and tertiary structures based on homology to known proteins. Secondary structure prediction methods include the methods proposed by Chou and Fasmann [70], and Garnier et al. [112]. Artificial neural networks [280, 287] and nearest neighbor methods [294, 295] are also used for this purpose. Tertiary structure prediction methods [349] are based on energy minimization, molecular dynamics, and stochastic searches of conformational space.

1.3.6 Molecular Design and Molecular Docking

When two molecules are in close proximity, it can be energetically favorable for them to bind together tightly. The molecular docking problem is the prediction of energy and physical configuration of binding between two molecules. A typical application is in drug design, in which one might dock a small molecule that is a described drug to an enzyme one wishes to target. For example, HIV protease is an enzyme in the AIDS virus that is essential to its replication. The chemical action of the protease takes place at a localized active site on its surface. HIV protease inhibitor drugs are small molecules that bind to the active site in HIV protease and stay there, so that the normal functioning of the enzyme is prevented. Docking software allows us to evaluate a drug design by predicting whether it will be successful in binding tightly to the active site in the enzyme. Based on the success of docking, and the resulting docked configuration, designers can refine the drug molecule [63, 188, 232, 374].

On the other hand, quantitative structure–activity relationship deals with establishing a mathematical correlation between calculated properties of molecules and their experimentally determined biological activity. These relationships may further

help in predicting the activity of new analogs that can be used as a drug for specific target [124, 154, 332].

1.3.7 Phylogenetic Trees for Studying Evolutionary Relationship

All species on earth undergo a slow transformation process called evolution. To explain the evolutionary history of today's species and how species relate to one another in terms of common ancestors, trees are constructed whose leaves represent the present-day species and intermediate nodes represent the hypothesized ancestors. These kind of labeled binary trees are called phylogenetic trees [163, 186, 191, 218, 308, 315]. Phylogenetic analysis is used to study the evolutionary relationship. Phylogenies are reconstructed based on comparisons between present-day objects. Given data for a set of objects, the phylogenetic tree reconstruction problem is to find the particular permutation of objects that optimize the given criteria. A number of algorithms are available to solve this problem [218, 308, 315].

1.3.8 Analysis of Microarray Expression Data

Microarray is one of the high throughput screening methods [281]. It measures the amount of mRNA in a sample that corresponds to a given gene or probe DNA sequence. Probe sequences are immobilized on a solid surface and allowed to hybridize with fluorescently-labeled target mRNA. The intensity of fluorescence of a spot is proportional to the amount of target sequence that has hybridized to that spot, and therefore, to the abundance of that mRNA sequence in the sample. Microarrays allow for identification of candidate genes involved in a given process based on variation between transcript levels for different conditions and shared expression patterns with genes of known function. A microarray data can be represented by a real-valued expression table [275]. A large amount of mRNA and miRNA profiling have been done and deposited in databases like Gene Expression Omnibus [94] and ArrayExpress [267]. Lot of works have been done using expression data to understand the activity of genes or nongenic elements like miRNA in several important cellular functions.

1.3.8.1 Clustering Genes or mRNAs

Clustering is one of the major tasks in gene expression data analysis [17, 157]. To understand gene function, gene regulation, cellular processes, and subtypes of cells, clustering techniques have proven to be helpful. The genes with similar expression patterns are likely to be involved in the same cellular processes, and a strong correlation of expression patterns between those genes indicates coregulation [95, 335].

Searching for common DNA sequences at the promoter regions of genes within the same cluster allows regulatory motifs specific to each gene cluster to be identified and cis-regulatory elements to be proposed [48, 335]. The inference of regulation through gene expression data clustering also gives rise to hypotheses regarding the mechanism of transcriptional regulatory network [87].

The conventional clustering methods such as hierarchical clustering [139], k-means algorithm [141], self-organizing map [334], principal component analysis [223], graph theoretical approaches [30, 36, 135, 310, 372, 373, 381, 384], model-based clustering [66, 103, 114, 145, 222, 224, 341, 352, 371, 382, 383], density-based approaches [156], fuzzy clustering algorithms [50, 83, 113], and rough–fuzzy clustering algorithms [212, 213] group coexpressed genes from microarray data. Different supervised gene clustering algorithms are also developed in [84, 136, 204, 205] to find coregulated gene clusters by incorporating the information of sample categories in gene clustering process. After clustering genes, a reduced set of genes can be selected for further analysis.

1.3.8.2 Clustering miRNAs

Recent genome wide surveys on noncoding RNAs have revealed that a substantial fraction of miRNAs is likely to form clusters. The genes of miRNAs are often organized in clusters in the genome. Expression analyses have showed strong positive correlations among the closely located miRNAs, indicating that they may be controlled by common regulatory elements. In fact, experimental evidence has demonstrated that clustered miRNA loci form an operon-like gene structure and that they are transcribed from common promoter [9]. Existence of coexpressed miRNAs is also demonstrated using expression profiling analysis in [28]. Several miRNA clusters have been experimentally shown by RT-PCR or Northern blotting [54, 183]. These findings suggest that members of a miRNA cluster, which are at a close proximity on a chromosome, are highly likely to be processed as cotranscribed units.

Expression data of miRNAs can be used to detect clusters of miRNAs as it is suggested that coexpressed miRNAs are cotranscribed, so they should have similar expression pattern. The complex miRNA–mRNA networks greatly increase the challenges of comprehending and interpreting the resulting mass of data. A first step toward addressing this challenge is the use of clustering techniques, which is essential in the pattern recognition process to reveal natural structures and identify interesting patterns in the underlying data [157]. Applying cluster techniques, miRNAs having similar cellular activities can be grouped together. In this background, several authors used hierarchical clustering in order to group miRNAs having similar function [96, 201, 354]. In [25], the self-organizing map has been used to cluster miRNA expression profile. Recently, Maji and Paul introduced rough–fuzzy clustering for identification of coexpressed miRNA clusters [212].

1.3.8.3 Selection of Genes or mRNAs

An important application of gene expression data in functional genomics is to classify samples according to their gene expression profiles, such as to classify cancer versus normal samples or to classify different types or subtypes of cancer [119]. However, among the large number of genes present in a microarray data set, only a small fraction of them is effective for classifying samples into different classes. Hence, one of the relevant and challenging problems in gene expression-based disease classification is the selection of effective genes from microarray data [326]. This is an important problem in machine learning and referred to as feature selection. In this regard, different feature selection methods have been used to select differentially expressed genes [6, 35, 119, 288, 291]. The small subset of genes is desirable in developing gene expression-based diagnostic tools for delivering precise, reliable, and interpretable results. With the gene selection results, the cost of biological experiment and decision can also be greatly reduced by analyzing only the marker genes.

1.3.8.4 Selection of miRNAs

Multiple reports have noted the utility of miRNAs for the diagnosis of cancer and other diseases [201]. The functions of miRNAs appear to be different in different cellular functions. Just as miRNA is involved in the normal functioning of eukaryotic cells, so has dysregulation of miRNA been associated with disease [184]. Different methods have been developed to identify potential miRNAs involved in a particular disease. In this background, the method called significance analysis of microarrays has been used to select potential miRNAs from expression data [151, 192, 238, 247, 274, 285]. Studies have also been conducted using various statistical tests like t-test and F-test for identifying differentially expressed miRNAs [129, 301, 355, 392]. The pattern recognition and machine learning techniques have been successfully used in [269, 270] to select differentially expressed miRNAs. The miRNA selection helps to infer about the miRNA–mRNA endogenous correlation associated in a disease.

1.3.8.5 Integration of mRNA and miRNA Expression Data

The high throughput techniques are used to generate a huge amount of mRNA and miRNA expression data. Individually, gene or mRNA expression data has been used to identify potential biomarkers for a wide ranges of diseases. However, gene or mRNA expression data alone often does not reflect robust molecular subtypes in many diseases. The small sample size and high number of features in the expression data put a challenge to extract meaningful information [275]. In order to generate a robust predictive model, few studies have been conducted by integrating different kinds of omics data [216, 221, 339]. While carrying out these types of integrated analyses, properties and scales have to be taken into account as well as the relations between different types of features.

The miRNA regulates gene expression at the posttranscriptional level. Hence, expression data of miRNA can provide information that is complementary to mRNA expression data. In this regard, data fusion could lead to improved classification accuracy [359]. It also helps to infer about the miRNA–mRNA endogenous correlation associated in a disease. Hence, combining miRNA and mRNA expression data, several methods have been developed that can clearly differentiate expression data into different types or subtypes and can reveal the correlation between miRNA and mRNA in a particular disease [90, 347].

1.3.8.6 Inference of Gene Regulatory Networks

One of the most challenging problems in the field of bioinformatics is inferring a gene regulatory network (GRN) from gene expression data [5]. An important and interesting question in biology, regarding the variation of gene expression levels, is how genes are regulated. Since almost all cells in a particular organism have an identical genome, differences in gene expression, and not the genome content, are responsible for cell differentiation during the life of the organism. For gene regulation, an important role is played by a type of proteins called transcription factors [308]. The transcription factors bind to specific parts of the DNA, called TFBS, which are specific, relatively short combinations of **A**, **T**, **C**, or **G**, and located in promoter regions. Transcription factors control gene expression by binding to the gene's promoter and either activating the gene or repressing it. They are gene products and therefore, in turn, can be controlled by other transcription factors. Transcription factors can control many genes, and some genes are controlled by combinations of transcription factors. Feedback loops are also possible.

Gene expression data can be used to infer regulatory relationships. This approach is known as reverse engineering of regulatory networks. Segal et al. [306] and Timothy et al. [340] highlighted that expression data can be used to make predictions about the transcriptional regulators for a given gene or sets of genes. Segal et al. [306] have developed a probabilistic model to identify modules of coregulated genes, their transcriptional regulators, and conditions that influence regulation. Timothy et al. [340] described a method to infer regulatory relationships which uses nonlinear differential equations to model regulatory networks. In this method, a model of connections between genes in a network is inferred from measurements of system dynamics, for example, response of genes and proteins to perturbations. Greenfield et al. [121] developed a hybrid method for incorporating structure priors into global regulatory networks inference. A new model for the GRNs has been developed in [346], which builds upon existing models by adding an epigenetic control layer. An approach for network inference by integrating expression plasticity into Shannon's mutual information is described in [356] for reconstruction of the GRNs. An integrated method has been developed for reconstructing the GRNs [88], utilizing both temporal information arriving from time-series gene expression profiles and topological properties of protein networks.

Reverse engineering based on differential equations and Bayesian networks was used by Cantone et al. [57] to identify regulatory interactions from time-series and steady-state expression data. A GRN inference algorithm from gene expression data based on differential equation model has been developed in [393]. Path consistency algorithm based on conditional mutual information has been employed for inferring the GRNs from gene expression data considering the nonlinear dependence and topological structure of the GRNs [390]. Liang and Wang [193] proposed a relevance network model for the GRN inference. Both mutual information and conditional mutual information have been used to determine the interactions between genes. Liang et al. [194] also developed a mutual information based REVerse Engineering ALgorithm, called REVEAL, for understanding interaction of genes. Later, Baladoni et al. [18] provided a new definition of mutual information using concepts from fuzzy set theory and extended the model on which the REVEAL algorithm for reverse engineering of the GRNs is based and they designed a new flexible version of it, called FuzzyREVEAL.

Microarrays have also been used in drug discovery [53], and its applications include basic research and target discovery, biomarker determination, pharmacology, toxicogenomics, target selectivity, development of prognostic tests, and disease-subclass determination. It is also used for gene set enrichment analysis [328]. Other potential bioinformatics tasks for biological problems are as follows: characterization of protein content and metabolic pathways between different genomes; identification and analysis of interacting proteins; characterization of repeats from genomes; gene mapping on chromosomes; analysis of genomic-scale censuses; assignment and prediction of gene products; large-scale analysis of gene expression levels; mapping expression data to sequence, structural, and biochemical data; development of digital libraries for automated bibliographical searches; development of knowledge bases of biological information from the literature; and development of DNA analysis methods in forensics [12, 77, 100, 137, 331, 353].

1.4 Pattern Recognition Perspective

Pattern recognition and machine learning tools and techniques have been widely used for analysis of biological data as classification, clustering, and feature selection are needed to analyze large biological data sets. Pattern recognition is the multidisciplinary research area that is concerned with the classification or description of objects. It aims to classify data or patterns based on either a prior knowledge or statistical information extracted from the data. Hence, in pattern recognition, mathematical, statistical, heuristic, and inductive techniques are utilized to execute the tasks like human being on computers [22, 85, 92, 108, 155, 209, 258, 260, 261, 263, 343].

At present, pattern recognition and machine learning provide the most fruitful framework for bioinformatics [20, 22, 209, 302, 377, 391]. They provide a wide range of linear and nonlinear, comprehensible and complex, predictive and descriptive, instance and rule-based models for different data mining tasks such as

dimensionality reduction, clustering, classification, and rule discovery. Also, the methods for modeling probabilistic and fuzzy uncertainties, vagueness, and incompleteness in the discovered patterns form a part of pattern recognition research. Another aspect that makes pattern recognition algorithms attractive for computational biology and bioinformatics is their capability of learning or induction. As opposed to many statistical techniques that require the user to have a hypothesis in mind first, pattern recognition algorithms and machine learning techniques automatically analyze the data and identify relationships among attributes and entities in the data to build models that allow domain experts to understand the relationship between the attributes and the class. Several data preprocessing tasks such as instance selection, data cleaning, dimensionality reduction, and handling missing data are also extensively studied in pattern recognition framework. Besides these, other data mining issues addressed by pattern recognition methodologies include handling of relational, sequential, and symbolic biological data, knowledge encoding and extraction, knowledge evaluation, and visualization.

Pattern recognition is at the core of data mining systems. However, pattern recognition and data mining are not equivalent considering their original definitions. There exists a gap between the requirements of a data mining system and the goals achieved by present-day pattern recognition algorithms. Development of new generation pattern recognition algorithms is expected to encompass more massive biological data sets involving diverse sources and types of data that will support mixed initiative data mining, where human experts collaborate with the computer to form hypotheses and test them.

1.4.1 Pattern Recognition

Pattern recognition is a two step procedure. The first step consists of learning the invariant and common properties of a set of samples characterizing a class. While in second step, it is decided whether a new sample is a possible member of the class or not, by noting that it has properties common to those of the set of samples. The task of pattern recognition can be described as a transformation from the measurement space $\mathscr{M}$ to the feature space $\mathscr{F}$ and finally to the decision space $\mathscr{D}$; that is,

$$\mathscr{M} \rightarrow \mathscr{F} \rightarrow \mathscr{D},$$

where the mapping $\delta : \mathscr{F} \rightarrow \mathscr{D}$ is the decision function, and the elements $d \in \mathscr{D}$ are termed as decisions [92, 209, 336].

A pattern recognition process can be decomposed into a series of few steps: data acquisition; data preprocessing; feature selection; and classification or clustering. The data acquisition phase includes gathering of data via a set of sensors depending on the environment within which the objects are to be classified. A raw data contains noise, so some preprocessing tasks such as noise reduction, filtering, encoding, and enhancement are applied on the collected data for extracting pattern vectors. The

dimension of the preprocessed data is then reduced by retaining or measuring only some characteristic features or properties. However, in a broader perspective, this stage significantly influences the entire recognition process. The last phase comprises the construction of classifier, in that a transformation relationship is established between features and classes [22, 92, 155, 209, 260, 336].

1.4.1.1 Data Acquisition and Preprocessing

Data acquisition is the process of gathering data via a set of sensors depending on the environment within which the objects are to be classified. Pattern recognition techniques are applicable in a wide domain, where the data may be qualitative, quantitative, or both; they may be numerical, linguistic, pictorial, or any combination thereof. Generally, the data structures that are used in bioinformatics are of two types: object data vectors such as microarray expression data and relational data such as DNA or protein sequences. Object data, sets of numerical vectors of m features, are represented as $X = \{x_1, \cdots, x_i, \cdots, x_n\}$, a set of n feature vectors in the m-dimensional measurement space $\Re^m$. The ith object observed in the process has vector x_i as its numerical representation; x_{ij} is the jth ($j = 1, \cdots, m$) feature associated with the ith object. On the other hand, relational data are a set of n^2 numerical relationships, say r_{ij}, between pairs of objects. In other words, r_{ij} represents the extent to which objects x_i and x_j are related in the sense of some binary relationship ρ. If the objects that are pairwise related by ρ are called $O = \{o_1, \cdots, o_i, \cdots, o_n\}$, then $\rho : O \times O \rightarrow \Re$.

After data acquisition, a number of data preprocessing techniques [134] are applied on the collected data for extracting pattern vectors. Today's real-world biological databases are highly susceptible to noisy, missing, and inconsistent data due to their typically huge size and their likely origin from multiple, heterogeneous sources. Low-quality data may lead to low-quality mining results. The methods for data preprocessing are organized into following three categories, namely, data cleaning, data integration, and transformation. While data cleaning can be applied to remove noise and correct inconsistency in the data, the data integration merges data from multiple sources into a coherent data store. In data transformation, the data are transformed into forms appropriate for mining. For example, normalization may improve the accuracy and efficiency of mining algorithms involving distance measurements.

1.4.1.2 Feature Selection and Extraction

The process of feature selection or extraction includes selection of a map by which a sample in an m-dimensional measurement space is transformed into a point in a d-dimensional feature space, where $d < m$ [85, 343]. Mathematically, it finds a mapping of the form $y = f(x)$, by which a sample $x = [x_1, \cdots, x_j, \cdots, x_m]$ in an m-dimensional measurement space $\mathscr{M}$ is transformed into an object $y = [y_1, \cdots, y_j, \cdots, y_d]$ in a d-dimensional feature space $\mathscr{F}$. The objective of feature

selection or extraction is two fold: to retain or generate the optimum salient characteristics necessary for the recognition process and to reduce the dimensionality of the measurement space so that effective and easily computable algorithms can be devised for efficient classification.

In feature selection or extraction, a suitable criterion is formulated first to evaluate the goodness of a feature set and then an optimal set is searched in terms of the criterion. Features having potential to maximize (respectively, minimize) interclass (respectively, intraclass) distances, are considered to have optimal saliencies. The criterion of a good feature is that it should be unchanging with any other possible variation within a class, while emphasizing differences that are important in discriminating between patterns of different types. The major mathematical measures so far devised for the estimation of feature quality are mostly statistical in nature, and can be broadly classified into two categories, namely, feature selection in the measurement space [44, 130, 159, 172] and feature selection in a transformed space [34, 58, 86, 149]. The techniques in the first category generally reduce the dimensionality of the measurement space by discarding redundant or least information carrying features. On the other hand, those in the second category utilize all the information contained in the measurement space to obtain a new transformed space, thereby mapping a higher dimensional pattern to a lower dimensional one. This is referred to as feature extraction [85, 92, 343].

The relationship between a feature selection algorithm and the inducer or classifier chosen to evaluate the usefulness of the feature selection process can take three main forms, namely, embedded, filter, and wrapper. In embedded scheme, the inducer or classifier has its own feature selection algorithm, either explicit or implicit. The methods to induce logical conjunctions [367] and traditional machine learning tools such as decision trees and artificial neural networks [227] are a few examples of the embedded technique. The filter schemes are independent of the induction algorithm. If the feature selection process takes place before the induction step, the former can be seen as a filter of nonrelevant features prior to induction. The filter methods evaluate the goodness of the feature subset looking only at the intrinsic characteristics of the data, based on the relationship of each single feature with the class label by the calculation of simple statistics computed from the empirical distribution [79, 80, 133]. In wrapper approach [172], a search is conducted in the feature space, evaluating the goodness of each feature subset by estimating the accuracy of the specific classifier to be used [159].

The generation procedure of candidate feature subsets can be categorized into individual feature ranking [79, 80, 133] and feature subset selection [44, 130]. The former measures the relevance of each feature to the class and selects the top-ranked ones, and is commonly used due to its simplicity, scalability, and good empirical success [130]. Recently, different deterministic heuristic search methods such as sequential forward selection, sequential backward selection, sequential floating forward selection, and sequential floating backward selection [279] and nondeterministic heuristic search methods such as simulated annealing [169], genetic algorithm [143], and tabu search [117] are also used in feature selection.

The feature extraction methods determine an appropriate subspace of dimension d from the original feature space of dimension m, either in a linear or a nonlinear way. The linear transforms such as principal component analysis (PCA) [92], independent component analysis [34, 58, 72, 182], linear discriminant analysis [109], and projection pursuit [104] have been widely used in pattern recognition for feature extraction and dimensionality reduction. On the other hand, kernel PCA [138, 300] and multidimensional scaling [45, 241, 296] are two examples of nonlinear feature extraction techniques. The artificial neural networks have also been used for feature extraction [78, 110, 173, 200].

1.4.1.3 Classification and Clustering

In classification and clustering, a feature space is partitioned into regions, where each region represents a category of input. Accordingly, it attempts to assign every data object in the entire feature space to one of the possible classes or clusters. In real life, the classes of samples are not properly described. Instead, a finite and usually smaller number of samples are available, which often provide partial information for optimal design of feature selection algorithm or classification or clustering system. Under such circumstances, it is assumed that these samples are representative of the classes or clusters. Such a set of typical patterns is called a training set. On the basis of the information gathered from the samples in the training set, the pattern recognition systems are designed, that is, the values of the parameters of various pattern recognition methods are decided.

A classification (respectively, clustering) scheme is designed using labeled (respectively, unlabeled) data. In supervised learning, an algorithm is developed using objects with known classifications, and later it is asked to classify an unknown object based on the information acquired by it during training. Supervised learning is used for classifying different objects. Some of the well-known classifiers are Bayesian classifier [92, 343], naive Bayesian classifier [92, 343], decision tree [49, 62, 69, 225, 231, 234, 282, 292, 309, 333], multilayer perceptron [138, 140, 197], radial basis function network [138, 299], support vector machine [52, 298, 299, 351], and k-nearest neighbor method [92, 343].

On the other hand, clustering is performed through unsupervised learning. In cluster analysis, a given data set is divided into a set of clusters in such a way that two objects from the same cluster are as similar as possible and the objects from different clusters are as dissimilar as possible. In effect, it tries to mimic the human ability to group similar objects into classes and categories. A number of clustering algorithms have been proposed to suit different requirements [41, 59, 92, 106, 152, 153, 160]. There are mainly two types of clustering approaches, namely, partitive clustering algorithms like k-means [199, 202], k-modes [146], PAM, CLARA [164], and CLARANS [97, 240]; and hierarchical methods like AGNES [81, 164], DIANA [164], Chameleon [162], ROCK [126], CURE [125], and BIRCH [389]. Aside from the above two categories, there are also two classes of clustering tasks that require special attention, namely, clustering high-dimensional data (for example, CLIQUE

[3] and PROCLUS [2]) and constraint-based clustering [344, 345]. Many tools and concepts of statistical physics have also been used in pattern recognition such as fractals [91, 168, 170], renormalization group [115], Ising models [144, 327], bond percolation [148], and Gibbs–Markov random fields [147].

1.4.2 Relevance of Soft Computing

One of the important problems in bioinformatics is uncertainty. Imprecision in computations and vagueness in class definition are some of the sources of this uncertainty. The uncertainty may also be probabilistic and fuzzy in nature. Pattern recognition, by its nature, admits many approaches, to provide the appropriate solution of a given problem. For any pattern recognition system, one needs to achieve robustness with respect to random noise and failure of components and to obtain output in real time. It is also desirable for the system to be adaptive to the changes in environment. Moreover, a system can be made artificially intelligent if it is able to emulate some aspects of the human reasoning system. The system should also be able to handle nonlinear and/or overlapping classes to tackle real-life problems, and generate soft and hard decisions to make the system flexible.

Soft computing and machine learning approaches to pattern recognition are attempts to achieve these goals. Artificial neural network, decision tree, genetic algorithms, fuzzy sets, and rough sets are used as the tools in these approaches. The challenge is, therefore, to devise powerful pattern recognition methodologies by symbiotically combining these tools for analyzing biological data in more efficient ways. The systems should have the capability of flexible information processing to deal with real-life ambiguous situations and to achieve tractability, robustness, and low-cost solutions [209]. Connectionist or artificial neural network based approaches to pattern recognition are attempts to achieve some of these goals because of their major characteristics such as adaptivity, robustness or ruggedness, speed and optimality [42, 65, 138, 197, 276]. They are also suitable in data rich environments and are typically used for extracting embedded knowledge in the form of rules, quantitative evaluation of these rules, clustering, self-organization, classification, and regression. They have an advantage, over other types of machine learning algorithms, for scaling [37, 56, 89, 107, 131, 198, 210, 357, 375, 376, 380]. Investigations have also been made in the area of pattern recognition using genetic algorithms [22, 266]. Like neural networks, genetic algorithms [118] are also based on powerful metaphors from the natural world. They mimic some of the processes observed in natural evolution, which include crossover, selection, and mutation, leading to a stepwise optimization of organisms.

The fuzzy set theoretic classification approach is developed based on the realization that a pattern may belong to more than one class, with varying degrees of class membership. Accordingly, fuzzy decision theoretic, fuzzy syntactic, and fuzzy neural approaches are developed [40, 51, 258, 261]. These approaches can handle uncertainties, arising from vague, incomplete, linguistic, and overlapping patterns

at various stages of pattern recognition systems [40, 161, 258, 385]. The fuzzy set theory has greater flexibility to capture various aspects of incompleteness, impreciseness, or imperfection in information about a situation as it is a generalization of the classical set theory [385]. The relevance of fuzzy set theory in the realm of pattern recognition is adequately justified in [39, 40, 161, 258, 386]. Fuzzy sets have been successfully applied in pattern classification [179], clustering [39, 40, 93, 161, 178, 217, 248, 258], and image processing [23, 105, 165, 206, 252, 253, 255, 257, 342, 370]. In addition to pattern recognition, fuzzy sets find widespread applications in solving different problems in association rule mining [16, 203, 361], fuzzy information storage and retrieval [132], functional dependency [47], data summarization [67, 181], web mining [177, 237], granular computing [26, 27], microarray data analysis [33, 83], and so forth.

The theory of rough sets [265, 271, 316] has gained popularity in modeling and propagating uncertainty. It also deals with vagueness and incompleteness. Hence, rough sets have emerged as a potential pattern recognition tool. The main idea of rough sets is to construct or synthesize approximations, in terms of upper and lower bounds of concepts, properties, or relations from the acquired data. Here, the information granules and reducts form the key notions. Information granules formalize the concept of finite precision representation of objects in real-life situations, and the reducts represent the core of an information system, both in terms of objects and features, in a granular universe. It also provides a mathematical framework to capture uncertainties associated with human cognition process [208]. It is turning out to be methodologically significant to the domains of artificial intelligence and cognitive sciences, especially in the representation of and reasoning with vague and/or imprecise knowledge, data classification, data analysis, machine learning, and knowledge discovery [246, 277, 278, 316]. This approach is relatively new as compared to connectionist and fuzzy set theoretic approaches. Rough set theory has been applied successfully to pattern classification [10, 32, 122, 123, 289, 314, 320, 322, 323], clustering [15, 82, 142, 196, 207, 208, 256], feature selection [71, 75, 211, 214, 319, 321, 323], microarray data analysis [68, 98, 211, 318, 324, 350], prediction of biological activity of molecules [210], and image processing [158, 235, 259, 363–365].

There have been several attempts over the last two decades to evolve new approaches to pattern recognition and deriving their hybrids by judiciously combining the merits of several techniques [249, 261] involving mainly fuzzy logic, artificial neural networks, genetic algorithms, and rough set theory, for developing an efficient new paradigm called soft computing [387]. Here integration is done in a cooperative, rather than a competitive, manner. The result is a more intelligent and robust system providing a human interpretable, low-cost, approximate solution, as compared to traditional techniques. Neuro-fuzzy approach is perhaps the most visible hybrid paradigm [51, 228–230, 254, 261], realized so far, in soft computing framework. Besides the generic advantages, the neuro-fuzzy approach provides the corresponding application specific merits [76, 111, 120, 330, 368, 388, 394]. Rough–fuzzy [209, 250, 265] and neuro-rough [68, 158, 264] hybridizations are also proving to be fruitful frameworks for modeling human perceptions and

providing means for computing with words. The rough–fuzzy computing provides a powerful mathematical framework to capture uncertainties associated with the data. Other hybridized models for pattern recognition and data mining include neuro-genetic [46, 215, 251, 293, 362], rough-genetic [43, 317, 369], fuzzy-genetic [14, 61, 64, 116, 180, 233], rough-neuro-genetic [167], rough-neuro-fuzzy [11, 24, 243, 244, 262], and neuro-fuzzy-genetic [187, 195, 283, 290, 307, 358] approaches.

1.5 Scope and Organization of the Book

This book has 11 chapters describing various theories, methodologies, and algorithms, along with extensive experimental results, addressing certain tasks of computational biology, and bioinformatics in pattern recognition paradigm with real-life applications. Various methodologies are described using information theoretic and soft computing approaches, for classification, feature selection, and clustering. The emphasis of the methodologies is given on handling both object and relational biological data sets that are large both in size and dimension, and involve classes that are overlapping, intractable, and/or having nonlinear boundaries. The effectiveness of the algorithms is demonstrated on different real-life biological data sets taken from varied domains of bioinformatics and medical imaging such as DNA and protein sequence data analysis, microarray data analysis, and medical imagery.

Chapter 2 presents the design of a hybrid learning model, termed as neural network tree (NNTree) for identification of splice-junction and protein coding region in DNA sequences. It incorporates the advantages of both decision tree and artificial neural network. An NNTree is a decision tree, where each nonterminal node contains a neural network. The idea is to use the framework of multilayer perceptron to design tree-structured pattern classifier. At each nonterminal node, the multilayer perceptron partitions the data set into m subsets; m being the number of classes in the data set present at that node. The NNTree is designed by splitting the nonterminal nodes of the tree by maximizing classification accuracy of the multilayer perceptron. In effect, it produces a reduced height m-ary tree. The versatility of this method is illustrated through its application in diverse fields. The effectiveness of the hybrid algorithm, along with a comparison with other related algorithms, is demonstrated on a set of benchmark data sets. Simulation results show that the NNTree achieves excellent performance in terms of classification accuracy, size of the tree, and classification time. Demonstrating its success in splice-junction and gene identification problems provides the effectiveness of this approach. Extensive experimental results establish that the NNTree classifier produces more accurate classifier than that have previously been obtained for a range of different sequence lengths, thereby indicating a cost-effective alternative in splice-junction and protein coding region identification problems.

The prediction of protein functional sites is an important issue in protein function studies and drug design. In order to apply the powerful kernel-based pattern recognition algorithms such as support vector machine to predict functional sites

in proteins, amino acids need encoding prior to input. In this regard, a new string kernel function, termed as the modified bio-basis function, is presented in Chap. 3. It maps a nonnumerical sequence space to a numerical feature space. The new string kernel function is developed based on the conventional bio-basis function and needs a bio-basis string as a support like conventional kernel function. The concept of zone of influence of bio-basis string is introduced in the new kernel function to take into account the influence of each bio-basis string in nonnumerical sequence space. An efficient method is described to select a set of bio-basis strings for the new kernel function, integrating the Fisher ratio, and a novel concept of degree of resemblance. The integration enables the method to select a reduced set of relevant and nonredundant bio-basis strings. Some quantitative indices are described for evaluating the quality of selected bio-basis strings. The effectiveness of the new string kernel function and bio-basis string selection method, along with a comparison with existing bio-basis function and related bio-basis string selection methods, is demonstrated on different protein data sets using the support vector machine.

Quantitative structure activity relationship (QSAR) is one of the important disciplines of computer-aided drug design that deals with the predictive modeling of properties of a molecule. In general, each QSAR data set is small in size with large number of features or descriptors. Among the large amount of descriptors present in the QSAR data set, only a small fraction of them is effective for performing the predictive modeling task. Chapter 4 presents a rough set-based feature selection algorithm to select a set of effective molecular descriptors from a given QSAR data set. The new algorithm selects the set of molecular descriptors by maximizing both relevance and significance of the descriptors. An important finding is that the new feature selection algorithm is shown to be effective in selecting relevant and significant molecular descriptors from the QSAR data set for predictive modeling. The performance of the new algorithm is studied using the R^2 statistic of support vector regression method. The effectiveness of the new algorithm, along with a comparison with existing algorithms, is demonstrated on several QSAR data sets.

Microarray technology is one of the important biotechnological means that allows to record the expression levels of thousands of genes simultaneously within a number of different samples. An important application of microarray gene expression data in functional genomics is to classify samples according to their gene expression profiles. Among the large amount of genes present in microarray gene expression data, only a small fraction of them is effective for performing a certain diagnostic test. In this regard, mutual information has been shown to be successful for selecting a set of relevant and nonredundant genes from microarray data. However, information theory offers many more measures such as the f-information measures that may be suitable for selection of genes from microarray gene expression data. Chapter 5 presents different f-information measures as the evaluation criteria for gene selection problem. To compute the gene–gene redundancy (respectively, gene-class relevance), these information measures calculate the divergence of the joint distribution of two genes' expression values (respectively, the expression values of a gene and the class labels of samples) from the joint distribution when two genes (respectively, the gene and class label) are considered to be completely independent. The performance of different

f-information measures is compared with that of mutual information based on the predictive accuracy of naive Bayes classifier, k-nearest neighbor rule, and support vector machine. An important finding is that some f-information measures are shown to be effective for selecting relevant and nonredundant genes from microarray data. The effectiveness of different f-information measures, along with a comparison with mutual information, is demonstrated on several cancer data sets.

One of the most important and challenging problems in functional genomics is how to select the disease genes. In Chap. 6, a computational method is reported to identify disease genes, judiciously integrating the information of gene expression profiles and the shortest path analysis of protein–protein interaction networks. While the gene expression profiles have been used to select differentially expressed genes as disease genes using mutual information based maximum relevance–maximum significance framework, the functional protein association network has been used to study the mechanism of diseases. Extensive experimental study on colorectal cancer establishes the fact that the genes identified by the integrated method have more colorectal cancer genes than the genes identified from the gene expression profiles alone, irrespective of any gene selection algorithm. Also, these genes have greater functional similarity with the reported colorectal cancer genes than the genes identified from the gene expression profiles alone. All these results indicate that the integrated method is quite promising and may become a useful tool for identifying disease genes.

The miRNAs regulate expression of a gene or protein. It has been observed that they play an important role in various cellular processes and thus help in carrying out normal functioning of a cell. However, dysregulation of miRNAs is found to be a major cause of a disease. Various studies have also shown the role of miRNAs in cancer and utility of miRNAs for the diagnosis of cancer. A large number of works have been conducted to identify differentially expressed miRNAs as unlike with mRNA expression, a modest number of miRNAs might be sufficient to classify human cancers. In this regard, Chap. 7 presents a new approach for selecting miRNAs from microarray expression data. It integrates the merit of rough set-based feature selection algorithm reported in Chap. 4, theory of $B.632+$ bootstrap error rate, and support vector machine. The effectiveness of the new approach, along with a comparison with other algorithms, is demonstrated on several miRNA data sets.

Clustering is one of the important analysis in functional genomics that discovers groups of coexpressed genes from microarray data. In Chap. 8, the application of a new partitive clustering algorithm, termed as rough-fuzzy c-means, is presented to discover coexpressed gene clusters. One of the major issues of rough-fuzzy c-means based microarray data clustering is how to select initial prototypes of different clusters. To overcome this limitation, a method is reported based on Pearson's correlation coefficient to select initial cluster centers. It enables the algorithm to converge to an optimum or near optimum solutions and helps to discover coexpressed gene clusters. A method is also presented based on cluster validity index to identify optimum values of different parameters of the initialization method and the clustering algorithm. The effectiveness of rough-fuzzy c-means algorithm, along with a comparison with other

clustering algorithms, is demonstrated on several yeast gene expression time-series data sets using different cluster validity indices and gene ontology based analysis.

In functional genomics, an important application of microarray data is to classify samples according to their gene expression profiles such as to classify cancer versus normal samples or to classify different types or subtypes of cancer. Hence, one of the major tasks with the gene expression data is to find groups of coregulated genes whose collective expression is strongly associated with the sample categories or response variables. In this regard, a supervised gene clustering algorithm is presented in Chap. 9 to find groups of genes. It directly incorporates the information of sample categories into the gene clustering process. A new quantitative measure, based on mutual information, is introduced that incorporates the information of sample categories to measure the similarity between attributes. The supervised gene clustering algorithm is based on measuring the similarity between genes using the new quantitative measure, whereby redundancy among the attributes is removed. The clusters are then refined incrementally based on sample categories. The performance of the new algorithm is compared with that of existing supervised and unsupervised gene clustering and gene selection algorithms based on the class separability index and the predictive accuracy of naive Bayes classifier, k-nearest neighbor rule, and support vector machine on several cancer and arthritis microarray data sets. The biological significance of the generated clusters is interpreted using the gene ontology. An important finding is that the supervised gene clustering algorithm is shown to be effective for identifying biologically significant gene clusters with excellent predictive capability.

The biclustering method is another important tool for analyzing gene expression data. It focuses on finding a subset of genes and a subset of experimental conditions that together exhibit coherent behavior. However, most of the existing biclustering algorithms find exclusive biclusters, which is inappropriate in the context of biology. Since biological processes are not independent of each other, many genes may participate in multiple different processes. Hence, nonexclusive biclustering algorithms are required for finding overlapping biclusters. In Chap. 10, a novel possibilistic biclustering algorithm is presented to find highly overlapping biclusters of larger volume with mean squared residue lower than a predefined threshold. It judiciously incorporates the concept of possibilistic clustering algorithm into biclustering framework. The integration enables efficient selection of highly overlapping coherent biclusters with mean squared residue lower than a given threshold. The detailed formulation of the new possibilistic biclustering algorithm, along with a mathematical analysis on the convergence property, is presented. Some quantitative indices are reported for evaluating the quality of generated biclusters. The effectiveness of the algorithm, along with a comparison with other algorithms, is demonstrated both qualitatively and quantitatively on yeast gene expression data set. In general, the new algorithm shows excellent performance at finding patterns in gene expression data.

Finally, Chap. 11 reports a robust thresholding technique for segmentation of brain MR images. It is based on the fuzzy thresholding techniques. Its aim is to threshold the gray level histogram of brain MR images by splitting the image histogram into multiple crisp subsets. The histogram of the given image is thresholded according to

the similarity between gray levels. The similarity is assessed through a second-order fuzzy measure such as fuzzy correlation, fuzzy entropy, and index of fuzziness. To calculate the second-order fuzzy measure, a weighted cooccurrence matrix is presented, which extracts the local information more accurately. Two quantitative indices are introduced to determine the multiple thresholds of the given histogram. The effectiveness of the algorithm, along with a comparison with standard thresholding techniques, is demonstrated on a set of brain MR images.

References

1. Adrahams JP, Breg M (2000) Prediction of RNA secondary structure including pseudoknoting by computer simulation. Nucl Acids Res 18:3035–3044
2. Agarwal CC, Procopiuc C, Wolf J, Yu PS, Park JS (1999) Fast algorithms for projected clustering. In: Proceedings of the ACM-SIGMOD international conference on management of data, Philadelphia, USA, pp 61–72
3. Agrawal R, Gehrke J, Gunopulos D, Raghavan P (1998) Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of the ACM-SIGMOD international conference on management of data, Seattle, WA, pp 94–105
4. Aho AV, Corasick M (1975) Efficient string matching: an aid to bibliographic search. Commun ACM 18(6):333–340
5. Akutsu T, Miyano S, Kuhara S (1999) Identification of genetic networks from a small number of gene expression patterns under the boolean network model. Proc Pac Symp Biocomput 99:17–28
6. Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceed Natl Acad Sci USA 96(12):6745–6750
7. Altschul SF, Gish W, Miller W, Myers E, Lipman DJ (1990) Basic local alignment search tool. Journal Mol Biol 215:403–410
8. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucl Acids Res 25(17):3389–3402
9. Altuvia Y, Landgraf P, Lithwick G, Elefant N, Pfeffer S, Aravin A, Brownstein MJ, Tuschl T, Margalit H (2005) Clustering and conservation patterns of human microRNAs. Nucl Acids Res 33:2697–2706
10. Ananthanarayana VS, Murty MN, Subramanian DK (2003) Tree structure for efficient data mining using rough sets. Patt Recogn Lett 24(6):851–862
11. Ang KK, Quek C (2006) Stock trading using RSPOP: a novel rough set-based neuro-fuzzy approach. IEEE Trans Neural Netw 17(5):1301–1315
12. Ansari HR, Raghava GP (2010) Identification of NAD interacting residues in proteins. BMC Bioinform 11:160
13. Arrigo P, Giuliano F, Damiani G (1991) Identification of a new motif on nucleic acid sequence data using Kohonen's self-organising map. Comput Appl Biosci 7(3):353–357
14. Ascia G, Catania V, Panno D (2006) An integrated fuzzy-GA approach for buffer management. IEEE Trans Fuzzy Syst 14(4):528–541
15. Asharaf S, Shevade SK, Murty MN (2005) Rough support vector clustering. Patt Recogn 38:1779–1783
16. Au WH, Chan KCC (1998) An effective algorithm for discovering fuzzy rules in relational databases. In: Proceedings of the IEEE international conference on fuzzy systems, pp 1314–1319

17. Au WH, Chan KCC, Wong AKC, Wang Y (2005) Attribute clustering for grouping, selection, and classification of gene expression data. IEEE/ACM Trans Comput Biol Bioinform 2(2):83–101
18. Badaloni S, Falda M, Massignan P, Sambo F (2009) Fuzzy mutual information for reverse engineering of gene regulatory networks. In: Proceedings of the international conference on fuzzy calculus, pp 25–30
19. Baldi P, Baisnee PF (2000) Sequence analysis by additive scales: DNA structure for sequences and repeats of all lengths. Bioinformatics 16:865–889
20. Baldi P, Brunak S (1998) Bioinformatics: The machine learning approach. MIT Press, Cambridge
21. Bandyopadhyay S (2005) An efficient technique for superfamily classification of amino acid sequences: feature extraction, fuzzy clustering and prototype selection. Fuzzy Sets Syst 152:5–16
22. Bandyopadhyay S, Pal SK (2007) Classification and learning using genetic algorithms: applications in bioinformatics and web intelligence. Springer, Heidelberg
23. Banerjee M, Kundu MK, Maji P (2009) Content-based image retrieval using visually significant point features. Fuzzy Sets Syst 160(23):3323–3341
24. Banerjee M, Mitra S, Pal SK (1998) Rough-fuzzy MLP: knowledge encoding and classification. IEEE Trans Neural Netw 9(6):1203–1216
25. Bargaje R, Hariharan M, Scaria V, Pillai B (2010) Consensus miRNA expression profiles derived from interplatform normalization of microarray data. RNA 16:16–25
26. Bargiela A, Pedrycz W (2003) Granular computing: an introduction. Kluwer Academic Publishers, Boston
27. Bargiela A, Pedrycz W (2008) Toward a theory of granular computing for human-centered information processing. IEEE Trans Fuzzy Syst 16(2):320–330
28. Baskerville S, Bartel DP (2005) Microarray profiling of microRNAs reveals frequent coexpression with neighboring miRNAs and host genes. RNA 11:241–247
29. Batenburg V, Gultyaev AP, Pleij CWA (1995) An APL-programmed genetic algorithm for the prediction of RNA secondary structure. J Theor Biol 174(3):269–280
30. Baya A, Granitto P (2011) Clustering gene expression data with a penalized graph-based metric. BMC Bioinform 12(1)
31. Bayley MJ, Jones G, Willett P, Williamson MP (1998) GENFOLD: a genetic algorithm for folding protein structures using NMR restraints. Protein Sci 7(2):491–499
32. Bazan JG, Skowron A, Synak P (1994) Discovery of decision rules from experimental data. In: Proceedings of the 3rd workshop on rough sets and soft computing, pp 526–533
33. Belacel N, Cuperlovic-Culf M, Laflamme M, Ouellette R (2004) Fuzzy J-means and VNS methods for clustering genes from microarray data. Bioinformatics 20(11):1690–1701
34. Bell AJ, Sejnowski TJ (1995) An information-maximization approach to blind separation and blind deconvolution. Neural Comput 7(6):1129–1159
35. Ben-Dor A, Bruhn L, Friedman N, Nachman I, Schummer M, Yakhini Z (2000) Tissue classification with gene expression profiles. J Comput Biol 7(3/4):559–584
36. Ben-Dor A, Shamir R, Yakhini Z (1999) Clustering gene expression patterns. J Comput Biol 6(3–4):281–297
37. Bengio Y, Buhmann JM, Embrechts M, Zurada JM (2000) Introduction to the special issue on neural networks for data mining and knowledge discovery. IEEE Trans Neural Netw 11: 545–549
38. Berry EA, Dalby AR, Yang ZR (2004) Reduced bio-basis function neural network for identification of protein phosphorylation sites: comparison with pattern recognition algorithms. Comput Biol Chem 28(1):75–85
39. Bezdek JC (1981) Pattern recognition with fuzzy objective function algorithm. Plenum Press, New York
40. Bezdek JC, Pal SK (1992) Fuzzy models for pattern recognition: methods that search for structures in data. IEEE Press, New York

41. Bhatia SK, Deogun JS (1998) Conceptual clustering in information retrieval. IEEE Trans Syst Man Cybern Part B: Cybern 28(3):427–436
42. Bian W, Xue X (2009) Subgradient-based neural networks for nonsmooth nonconvex optimization problems. IEEE Trans Neural Netw 20(6):1024–1038
43. Bjorvand AT, Komorowski J (1997) Practical applications of genetic algorithms for efficient reduct computation. In: Proceedings of the 15th IMACS world congress on scientific computation, modeling and applied mathematics, vol 4, pp 601–606
44. Blum AL, Langley P (1997) Selection of relevant features and examples in machine learning. Artif Intell 97:245–271
45. Borg I, Groenen P (1997) Modern multidimensional scaling. Springer, Berlin
46. Bornholdt S, Graudenz D (1992) General asymmetric neural networks and structure design by genetic algorithms. Neural Netw 5:327–334
47. Bosc P, Pivert O, Ughetto L (1999) Database mining for the discovery of extended functional dependencies. In: Proceedings of the 18th international conference of the North American Fuzzy Information Processing Society, IEEE Press, Piscataway, NJ, New York, USA, pp 580–584
48. Brazma A, Vilo J (2000) Minireview: gene expression data analysis. Fed Eur Biochem Soc Lett 480(1):17–24
49. Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classif Regres Trees. Wadsworth, Belmont
50. Brintha SJ, Bhuvaneswari V (2012) Clustering microarray gene expression data using type 2 fuzzy logic. In: Proceedings of the 3rd national conference on emerging trends and applications in computer, science, pp 147–151
51. Bunke H, Kandel A (eds) (2001) Neuro-fuzzy pattern recognition. World Scientific, Singapore
52. Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Data Mining Knowl Discov 2(2):121–167
53. Butte A (2002) The use and analysis of microarray data. Nature Rev Drug Discov 1(12):951–960
54. Cai X, Hagedorn CH, Cullen BR (2004) Human microRNAs are processed from capped, polyadenylated transcripts that can also function as mRNAs. RNA 10:1957–1966
55. Cai YD, Chou KC (1998) Artificial neural network model for predicting HIV protease cleavage sites in protein. Adv Eng Softw 29(2):119–128
56. Cai YD, Liu XJ, Xu XB, Chou KC (2002) Support vector machines for predicting the specificity of galNAc-transferase. Peptides 23:205–208
57. Cantone I, Marucci L, Iorio F, Ricci MA, Belcastro V, Bansal M, Santini S, di Bernardo M, di Bernardo D, Cosma MP (2009) A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. Cell 137(1):172–181
58. Cardoso J (1998) Blind signal separation: statistical principles. Proc IEEE 86:2009–2025
59. Carpineto C, Romano G (1996) A lattice conceptual clustering system and its application to browsing retrieval. Mach Learn 24(2):95–122
60. Carpio CAD (1996) A parallel genetic algorithm for polypeptide three dimensional structure prediction: a transputer implementation. J Chem Inform Comput Sci 36(2):258–269
61. Casillas J, Carse B, Bull L (2007) Fuzzy-XCS: a Michigan genetic fuzzy system. IEEE Trans Fuzzy Syst 15(4):536–550
62. Chandrashekharan B (1986) From numbers to symbols to knowledge structures: pattern recognition and artificial intelligence perspectives on the classification task, vol 2. Elsevier Science, Amsterdam
63. Chen C, Wang LH, Kao C, Ouhyoung M, Chen W (1998) Molecular binding in structured-based drug design: a case study of the population-based annealing genetic algorithms. In: Proceedings of the IEEE international conference on tools with, artificial intelligence, pp 328–335
64. Chen CH, Tseng VS, Hong TP (2008) Cluster-based evaluation in fuzzy-genetic data mining. IEEE Trans Fuzzy Syst 16(1):249–262

65. Chen H, Yao X (2009) Regularized negative correlation learning for neural network ensembles. IEEE Trans Neural Netw 20(12):1962–1979
66. Chen L, Jiang Q, Wang S (2012) Model-based method for projective clustering. IEEE Trans Knowl Data Eng 24(7):1291–1305
67. Chiang D, Chow LR, Wang Y (2000) Mining time series data by a fuzzy linguistic summary system. Fuzzy Sets Syst 112:419–432
68. Chiang JH, Ho SH (2008) A combination of rough-based feature selection and RBF neural network for classification using gene expression data. IEEE Trans NanoBiosci 7(1):91–99
69. Chou PA (1991) Optimal partitioning for classification and regression trees. IEEE Trans Patt Anal Mach Intell 13(4):340–354
70. Chou PY, Fasmann GD (1978) Prediction of the secondary structure of proteins from their amino acid sequence. Adv Enzymol Relat Areas Mol Biol 47:145–148
71. Chouchoulas A, Shen Q (2001) Rough set-aided keyword reduction for text categorisation. Appl Artif Intell 15(9):843–873
72. Comon P (1994) Independent component analysis, a new concept? Signal Process 36(3):287–314
73. Contreras-Moreira B, Fitzjohn PW, Offman M, Smith GR, Bates PA (2003) Novel use of genetic algorithm for protein structure prediction: searching template and sequence alignment space. Proteins 53(6):424–429
74. Cooper L, Corne D, Crabbe M (2003) Use of a novel Hill-climbing genetic algorithm in protein folding simulations. Comput Biol Chem 27(6):575–580
75. Cornelis C, Jensen R, Martin GH, Slezak D (2010) Attribute selection with fuzzy decision reducts. Inform Sci 180:209–224
76. Cpalka K (2009) A new method for design and reduction of neuro-fuzzy classification systems. IEEE Trans Neural Netw 20(4):701–714
77. Crollius HR, Jaillon O, Dasilva C, Ozouf-Costaz C, Fizames C, Fischer C, Bouneau L, Billault A, Quetier F, Saurin W, Bernot A, Weissenbach J (2000) Characterization and repeat analysis of the compact genome of the freshwater pufferfish tetraodon nigroviridis. Genome Res 10:939–949
78. Cun YL, Boser B, Denker JS, Henderson D, Horward RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1:541–551
79. Dash M, Liu H (1997) Feature selection for classification. Intell Data Anal 1:131–156
80. Dash M, Liu H (2003) Consistency-based search in feature selection. Artif Intell 151:155–176
81. Day WHE, Edelsbrunner H (1984) Efficient algorithms for agglomerative hierarchical clustering methods. J Classif 1(1):7–24
82. De SK (2004) A rough set theoretic approach to clustering. Fundam Inform 62(3–4):409–417
83. Dembele D, Kastner P (2003) Fuzzy C-means method for clustering microarray data. Bioinformatics 19(8):973–980
84. Dettling M, Buhlmann P (2002) Supervised clustering of genes. Genome Biol 3(12):1–15
85. Devijver PA, Kittler J (1982) Pattern recognition: a statistical approach. Prentice Hall, Englewood Cliffs
86. Devroye L, Gyorfi L, Lugosi G (1996) A probabilistic theory of pattern recognition. Springer, Berlin
87. D'haeseleer P, Wen X, Fuhrman S, Somogyi R (1998) Mining the gene expression matrix: inferring gene relationships from large scale gene expression data. In: Proceedings of the 2nd international workshop on information processing in cell and tissues, pp 203–212
88. Dimitrakopoulos G, Sgarbas K, Dimitrakopoulou K, Dragomir A, Bezerianos A, Maraziotis IA () Multi-scale modeling of gene regulatory networks via integration of temporal and topological biological data. In: Proceedings of the annual international conference of the IEEE Engineering in Medicine and Biology Society, pp 1242–1245
89. Ding CHQ, Dubchak I (2001) Multi-class protein fold recognition using support vector machines and neural networks. Bioinformatics 17:349–358
90. Dong H, Siu H, Luo L, Fang X, Jin L, Xiong M (2010) Investigation gene and microRNA expression in glioblastoma. BMC Genomics 11(Suppl 3):S16

91. Dubuisson MP, Dubes RC (1994) Efficacy of fractal features in segmenting images of natural textures. Patt Recogn Lett 15:419–431
92. Duda RO, Hart PE, Stork DG (1999) Pattern classification and scene analysis. Wiley, New York
93. Dunn JC (1974) A fuzzy relative of the ISODATA process and its use in detecting compact, well-separated clusters. J Cybern 3:32–57
94. Edgar R, Domrachev M, Lash AE (2002) Gene expression omnibus: NCBI gene expression and hybridization array data repository. Nucl Acids Res 30(1):207–210
95. Eisen MB, Spellman PT, Brown PO, Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci USA 95(25):14863–14868
96. Enerly E, Steinfeld I, Kleivi K, Leivonen SK, Aure MR, Russnes HG, Ronneberg JA, Johnsen H, Navon R, Rodland E, Makela R, Naume B, Perala M, Kallioniemi O, Kristensen VN, Yakhini Z, Dale ALB (2011) miRNA–mRNA integrated analysis reveals roles for miRNAs in primary breast tumors. PLoS One 6(2)
97. Ester M, Kriegel HP, Xu X (1995) Knowledge discovery in large spatial databases: focusing techniques for efficient class identification. In: Proceedings of the international symposium on large spatial databases, Portland, ME, pp 67–82
98. Fang J, Busse JWG (2006) Mining of MicroRNA expression data—a rough set approach. In: Proceedings of the 1st international conference on rough sets and knowledge technology. Springer, Berlin, pp 758–765
99. Farber R, Lapedes A, Sirotkin K (1992) Determination of eucaryotic protein coding regions using neural networks and information theory. J Mol Biol 226(2):471–479
100. Feng X, Mouttaki H, Lin L, Huang R, Wu B, Hemme CL, He Z, Zhang B, Hicks LM, Xu J, Zhou J, Tang YJ (2009) Characterization of the central metabolic pathways in thermoanaerobacter sp. strain X514 via isotopomer-assisted metabolite analysis. Appl Environ Microbiol 75(15):5001–5008
101. Fickett J (1982) Recognition of protein coding regions in DNA sequences. Nucl Acids Res 10(17):5303–5318
102. Fickett J, Tung CS (1992) Assessment of protein coding measures. Nucl Acids Res 20(24):6441–6450
103. Fraley C, Raftery AE (1998) How many clusters? Which clustering method? Answers via model-based cluster analysis. Comput J 41(8):578–588
104. Friedman JH (1987) Exploratory projection pursuit. J Am Stat Assoc 82(397):249–266
105. Frigui H (1999) Adaptive image retrieval using the fuzzy integral. In: Proceedings of the 18th international conference of the North American Fuzzy Information Processing Society, IEEE Press, Piscataway, NJ, New York, pp 575–579
106. Frigui H, Krishnapuram R (1999) A robust competitive clustering algorithm with applications in computer vision. IEEE Trans Patt Anal Mach Intell 21(5):450–465
107. Frolov AA, Husek D, Polyakov PY (2009) Recurrent-neural-network-based boolean factor analysis and its application to word clustering. IEEE Trans Neural Netw 20(7):1073–1086
108. Fu KS (1982) Syntactic pattern recognition and application. Prentice-Hall, Englewood Cliffs
109. Fukunaga K (1990) Introduction to statistical pattern recognition. Academic Press, New York
110. Fukushima K, Miyako S, Ito T (1983) Neocognitron: A Neural Network Model for a Mechanism of Visual Pattern Recognition. IEEE Trans Syst Man Cybern 13:826–834
111. Gajate A, Haber RE, Vega PI, Alique JR (2010) A transductive neuro-fuzzy controller: application to a drilling process. IEEE Trans Neural Netw 21(7):1158–1167
112. Garnier J, Gibrat JF, Robson B (1996) GOR method for predicting protein secondary structure from amino acid sequence. Methods Enzymol 266:540–553
113. Gasch AP, Eisen MB (2002) Exploring the conditional coregulation of yeast gene expression through fuzzy K-means clustering. Genome Biol 3(11):1–22
114. Ghosh D, Chinnaiyan AM (2002) Mixture modelling of gene expression data from microarray experiments. Bioinformatics 18:275–286
115. Gidas B (1989) A renormalization group approach to image processing letters. IEEE Trans Patt Anal Mach Intell 11(2):164–180

116. Giordano V, Naso D, Turchiano B (2006) Combining genetic algorithms and Lyapunov-based adaptation for online design of fuzzy controllers. IEEE Trans Syst Man Cybern Part B: Cybern 36(5):1118–1127
117. Glover F, Laguna M (1999) Tabu search. Kluwer Academic Publishers, Boston
118. Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading
119. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, Lander ES (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286(5439):531–537
120. Goncalves LB, Vellasco MMBR, Pacheco MAC, de Souza FJ (2006) Inverted hierarchical neuro-fuzzy BSP system: a novel neuro-fuzzy model for pattern classification and rule extraction in databases. IEEE Trans Syst Man Cybern Part C: Appl Rev 36(2):236–248
121. Greenfield A, Hafemeister C, Bonneau R (2013) Robust data-driven incorporation of prior knowledge into the inference of dynamic regulatory networks. Bioinformatics 29(8):1060–1067
122. Grzymala-Busse JW (1992) LERS—a system for learning from examples based on rough sets. In: Slowinski R (ed) Intelligent decision support, vol 11. Springer, The Netherlands, pp 3–18
123. Grzymala-Busse JW (1997) A new version of the rule induction system LERS. Fundam Inform 31(1):27–39
124. Guha R, Jurs PC (2004) Development of linear, ensemble, and nonlinear models for the prediction and interpretation of the biological activity of a set of PDGFR inhibitors. J Chem Inform Comput Sci 44:2179–2189
125. Guha S, Rastogi R, Shim K (1998) CURE: an efficient clustering algorithm for large databases. In: Proceedings of the ACM-SIGMOD international conference on management of data, Seattle, WA, pp 73–84
126. Guha S, Rastogi R, Shim K (19991) ROCK: a robust clustering algorithm for categorical attributes. In: Proceedings of the international conference on data engineering, Sydney, Australia, pp 512–521
127. Gultyaev AP, Batenburg V, Pleij CWA (1995) The computer simulation of RNA folding pathways using a genetic algorithm. J Mol Biol 250(1):37–51
128. Gunn JR (1997) Sampling protein conformations using segment libraries and a genetic algorithm. J Chem Phys 106:4270–4281
129. Guo J, Miao Y, Xiao B, Huan R, Jiang Z, Meng D, Wang Y (2009) Differential expression of microRNA species in human gastric cancer versus non-tumorous tissues. J Gastroenterol Hepatol 24:652–657
130. Guyon I (2003) Elisseeff: an introduction to variable and feature selection. J Mach Learn Res 3:1157–1182
131. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46(1–3):389–422
132. Hale J, Shenoi S (1996) Analyzing FD inference in relational databases. Data Knowl Eng 18:167–183
133. Hall MA (2000) Correlation-based feature selection for discrete and numeric class machine learning. In: Proceedings of the 17th international conference on machine learning, pp 359–366
134. Han J, Kamber M (2001) Data mining: concepts and techniques. Morgan Kaufmann Publishers, San Francisco
135. Hartuv E, Shamir R (2000) A clustering algorithm based on graph connectivity. Inf Process Lett 76(4–6):175–181
136. Hastie T, Tibshirani R, Botstein D, Brown P (2001) Supervised harvesting of expression trees. Genome Biol 1:1–12
137. Hawkins T, Chitale M, Luban S, Kihara D (2009) PFP: automated prediction of gene ontology functional annotations with confidence scores using protein sequence da. Proteins 74(3):566–582

138. Haykin S (1998) Neural networks: a comprehensive foundation, 2nd edn. Prentice Hall, Upper Saddle River
139. Herrero J, Valencia A, Dopazo J (2001) A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics 17:126–136
140. Hertz J, Krogh A, Palmer RG (1991) Introduction to the theory of neural computation. Santa Fe institute studies in the sciences of complexity. Addison Wesley, New York
141. Heyer LJ, Kruglyak S, Yooseph S (1999) Exploring expression data: identification and analysis of coexpressed genes. Genome Res 9(11):1106–1115
142. Hirano S, Tsumoto S (2003) An indiscernibility-based clustering method with iterative refinement of equivalence relations: rough clustering. J Adv Comput Intell Intell Inf 7(2):169–177
143. Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor
144. Honda Y, Horiguchi T, Miya M (1997) Restoration of digital images of the alphabet by using Ising models. Phys Lett A 227(5):319–324
145. Hong Y, Kwong S, Wang H, Ren Q, Chang Y (2008) Probabilistic and graphical model based genetic algorithm driven clustering with instance-level constraints. In: Proceedings of IEEE congress on evolutionary computation: IEEE World congress on computational intelligence, pp 322–329
146. Huang Z (1998) Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining Knowl Discov 2:283–304
147. Hussain I, Reed T (1995) A bond percolation-based Gibbs-Markov random fields for image segmentation. IEEE Signal Process Lett 2(8):145–147
148. Hussain I, Reed T (1997) A bond percolation-based model for image segmentation. IEEE Trans Patt Anal Mach Intell 6(12):1698–1704
149. Hyvarlnen A, Oja R (1997) A fast fixed-point algorithm for independent component analysis. Neural Comput 9(7):1483–1492
150. Iijima H, Naito Y (1994) Incremental prediction of the side-chain conformation of proteins by a genetic algorithm. In: Proceedings of the 1st IEEE conference on evolutionary computation, IEEE world congress on computational intelligence, vol 1, pp 362–367
151. Iorio MV, Visone R, Leva GD, Donati V, Petrocca F, Casalini P, Taccioli C, Volinia S, Liu CG, Alder H, Calin GA, Menard S, Croce CM (2007) MicroRNA signatures in human ovarian cancer. Cancer Res 67(18):8699–8707
152. Jain AK, Dubes RC (1988) Algorithms for clustering data. Prentice Hall, Englewood Cliffs
153. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv 31(3):264–323
154. Jain AN, Koile K, Chapman D (1994) Compass: predicting biological activities from molecular surface properties. performance comparisons on a steroid benchmark. J Med Chem 37:2315–2327
155. Jensen R, Shen Q (2008) Computational intelligence and feature selection: rough and fuzzy approaches. Wiley-IEEE Press, New York
156. Jiang D, Pei J, Zhang A (2003) DHC: a density-based hierarchical clustering method for time-series gene expression data. In: Proceedings of the 3rd IEEE international symposium on bioinformatics and bioengineering, pp 393–400
157. Jiang D, Tang C, Zhang A (2004) Cluster analysis for gene expression data: a survey. IEEE Trans Knowl Data Eng 16(11):1370–1386
158. Jiang J, Yang D, Wei H (2008) Image segmentation based on rough set theory and neural networks. In: Proceedings of the 5th international conference on visual information, engineering, pp 361–365
159. John GH, Kohavi R, Pfleger K (1994) Irrelevant features and the subset selection problem. In: Kaufmann M (ed) Proceedings of the 11th international conference on machine learning, New Brunswick, NJ, pp 121–129
160. Judd D, Mckinley P, Jain AK (1998) Large-scale parallel data clustering. IEEE Trans Patt Anal Mach Intell 20(8):871–876
161. Kandel A (1982) Fuzzy techniques in pattern recognition. Wiley Interscience, New York

162. Karypis G, Han EH, Kumar V (1999) Chameleon: a hierarchical clustering algorithm using dynamic modeling. Computer 32(8):68–75
163. Katoh K, Kuma K, Miyata T (2001) Genetic algorithm-based maximum likelihood analysis for molecular phylogeny. J Mol Evol 53(4):477–484
164. Kaufmann L, Rousseeuw PJ (1990) Finding groups in data: an introduction to cluster analysis. Wiley, New York
165. Kerre EE, Nachtegael M (eds) (2010) Fuzzy techniques in image processing. Physica-Verlag, Heidelberg
166. Khimasia M, Covency P (1997) Protein structure prediction as a hard optimization problem: the genetic algorithm approach. Mol Simul 19:205–226
167. Kiem H, Phuc D (1999) Using rough genetic and Kohonen's neural network for conceptual cluster discovery in data mining. In: Proceedings of the 7th international conference on rough sets. Fuzzy sets, data mining, and granular computing, Yamaguchi, Japan, pp 448–452
168. Killer JM, Chen SS, Crownover RM (1993) On the calculation of fractal features from images. IEEE Trans Patt Anal Mach Intell 15(10):1087–1090
169. Kirkpatrick S, Gelatt CDJ, Vecchi MP (1983) Optimization by simmulated annealing. Science 220:671–680
170. Klanderman GA, Huttenlocher DP, Rucklidge WJ (1993) Comparing images using the Hausdroff distance. IEEE Trans Patt Anal Mach Intell 15(9):850–863
171. Knudsen S (1999) Promoter 2.0: for the recognition pol II promoter sequences. Bioinformatics 15:356–361
172. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97(1–2):273–324
173. Kohonen T (2001) Self-organizing maps. Springer, Berlin
174. Konig R, Dandekar T (1999) Improving genetic algorithms for protein folding simulations by systematic crossover. BioSystems 50:17–25
175. Krasnogor N, Hart WE, Smith J, Pelta DA (1999) Protein structure prediction with evolutionary algorithms. In: Banzhaf W, Daida J, Eiben AE, Garzon MH, Honavar V, Jakiela M, Smith RE (eds) Proceedings of the international conference on genetic and evolutionary computation, vol 2, pp 1596–1601
176. Krasnogor N, Pelta D, Lopez PEM, de la Canal E (1998) Genetic algorithms for the protein folding problem: a critical view. In: Fyfe C, Alpaydin E (eds) Proceedings of the engineering of intelligent systems, pp 353–360
177. Krishnapuram R, Joshi A, Nasraoui O, Yi L (2001) Low complexity fuzzy relational clustering algorithms for web mining. IEEE Trans Fuzzy Syst 9:595–607
178. Krishnapuram R, Keller JM (1993) A possibilistic approach to clustering. IEEE Trans Fuzzy Syst 1(2):98–110
179. Kuncheva LI (2010) Fuzzy classifier design. Physica-Verlag, Heidelberg
180. Lee CS, Guo SM, Hsu CY (2005) Genetic-based fuzzy image filter and its application to image processing. IEEE Trans Syst Man Cybern Part B: Cybern 35(4):694–711
181. Lee DH, Kim MH (1997) Database Summarization Using Fuzzy ISA Hierarchies. IEEE Trans Syst Man Cybern Part B: Cybern 27:68–78
182. Lee TW (1993) Independent component analysis. Kluwer Academic Publishers, Dordrech
183. Lee Y, Kim M, Han J, Yeom KH, Lee S, Baek SH, Kim VN (2004) MicroRNA genes are transcribed by RNA polymerase II. EMBO J 23:4051–4060
184. Lehmann U, Streichert T, Otto B, Albat C, Hasemeier B, Christgen H, Schipper E, Hille U, Kreipe HH, Langer F (2010) Identification of differentially expressed microRNAs in human male breast cancer. BMC Bioinform 10(1–9)
185. Lehninger A, Nelson DL, Cox MM (2008) Lehninger principles of biochemistry, 5th edn. W. H. Freeman, New York
186. Lemmon AR, Milinkovitch MC (2002) The metapopulation genetic algorithm: an efficient solution for the problem of large phylogeny estimation. Proc Natl Acad Sci USA 99(16):10516–10521
187. Leng G, McGinnity TM, Prasad G (2006) Design for self-organizing fuzzy neural networks based on genetic algorithms. IEEE Trans Fuzzy Syst 14(6):755–766

188. Lesk AM (2002) Introduction to bioinformatics. Oxford University Press, London
189. LevItsky VG, Katokhin AV (2003) Recognition of eukaryotic promoters using a genetic algorithm based on iterative discriminant analysis. Sillico Biol 3(1–2):81–87
190. Lewin B (2003) Genes VIII. Benjamin Cummings
191. Lewis PO (1998) A genetic algorithm for maximum likelihood phylogeny inference using nucleotide sequence data. Mol Biol Evol 15(3):277–283
192. Li S, Chen X, Zhang H, Liang X, Xiang Y, Yu C, Zen K, Li Y, Zhang CY (2009) Differential expression of microRNAs in mouse liver under aberrant energy metabolic status. J Lipid Res 50:1756–1765
193. Liang KC, Wang X (2008) Gene regulatory network reconstruction using conditional mutual information. EURASIP J Bioinform Syst Biol 2008(1)
194. Liang S, Fuhrman S, Somogyi R (1998) REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In: Proceedings of the pacific symposium on biocomputing, pp 18–29
195. Liao GC, Tsao TP (2006) Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting. IEEE Trans Evol Comput 10(3):330–340
196. Lingras P, West C (2004) Interval set clustering of web users with rough K-means. J Intell Inf Syst 23(1):5–16
197. Lippmann R (1987) An introduction to computing with neural nets. IEEE Acoust Speech Signal Process Mag 4(2):4–22
198. Liu J, Gough J, Rost B (2006) Distinguishing protein-coding from non-coding RNAs through support vector machines. PLoS Genet 2(4):e29
199. Lloyd SP (1982) Least squares quantization in PCM. IEEE Trans Inf Theor 28:128–137 (Original version:Technical Report, Bell Labs, 1957)
200. Lowe D, Webb AR (1991) Optimized feature extraction and the Bayes decision in feed-forward classifier networks. IEEE Trans Patt Anal Mach Intell 13(4):264–355
201. Lu J, Getz G, Miska EA, Saavedra EA, Lamb J, Peck D, Cordero AS, Ebert BL, Mak RH, Ferrando AA, Downing JR, Jacks T, Horvitz HR, Golub TR (2005) MicroRNA expression profiles classify human cancers. Nat Lett 435(9):834–838
202. MacQueen J (1967) Some methods for classification and analysis of multivariate observation. In: Proceedings of the 5th Berkeley symposium on mathematical statistics and probability, vol 1, pp 281–297
203. Maeda A, Ashida H, Taniguchi Y, Takahashi Y (1995) Data mining system using fuzzy rule induction. In: Proceedings of the IEEE international conference on fuzzy systems, pp 45–46
204. Maji P (2011) Fuzzy-rough supervised attribute clustering algorithm and classification of microarray data. IEEE Trans Syst Man Cybern Part B: Cybern 41(1):222–233
205. Maji P, Das C (2012) Relevant and significant supervised gene clusters for microarray cancer classification. IEEE Trans NanoBiosci 11(2):161–168
206. Maji P, Kundu MK, Chanda B (2008) Second order fuzzy measure and weighted co-occurrence matrix for segmentation of brain MR images. Fundam Inform 88(1–2):161–176
207. Maji P, Pal SK (2007) RFCM:a hybrid clustering algorithm using rough and fuzzy sets. Fundam Inform 80(4):475–496
208. Maji P, Pal SK (2007) Rough set based generalized fuzzy C-means algorithm and quantitative indices. IEEE Trans Syst Man Cybern Part B: Cybern 37(6):1529–1540
209. Maji P, Pal SK (2012) Rough-fuzzy pattern recognition: applications in bioinformatics and medical imaging. Wiley-IEEE Computer Society Press, New York
210. Maji P, Paul S (2010) Rough sets for selection of molecular descriptors to predict biological activity of molecules. IEEE Trans Syst Man Cybern Part C: Appl Rev 40(6):639–648
211. Maji P, Paul S (2011) Rough set based maximum relevance-maximum significance criterion and gene selection from microarray data. Int J Approx Reason 52(3):408–426
212. Maji P, Paul S (2013) Robust rough-fuzzy C-means algorithm: Design and applications in coding and non-coding RNA expression data clustering. Fundam Inform 124:153–174

213. Maji P, Paul S (2013) Rough-fuzzy clustering for grouping functionally similar genes from microarray data. IEEE/ACM Trans Comput Biol Bioinform 10(2):286–299
214. Maji P, Paul S (2013) Rough set-based feature selection: criteria of max-dependency, max-relevance, and max-significance. In: Skowron A, Suraj Z (eds) Rough sets and intelligent systems—Professor Zdzis+aw Pawlak in memoriam, vol 43, pp 393–418
215. Maniezzo V (1994) Genetic evolution of the topology and weight distribution of neural networks. IEEE Trans Neural Netw 5:39–53
216. Martin L, Anguita A, Maojo V, Crespo J (2010) Integration of omics data for cancer research. In: Cho WCS (ed) An omics perspective on cancer research. Springer, The Netherlands, pp 249–266
217. Masulli F, Rovetta S (2006) Soft transition from probabilistic to possibilistic fuzzy clustering. IEEE Trans Fuzzy Syst 14(4):516–527
218. Matsuda H (1995) Protein phylogenetic inference using maximum likelihood with a genetic algorithm. In: Proceedings of the pacific symposium on biocomputing, pp 512–523
219. Maulik U, Bandyopadhyay S (2003) Fuzzy partitioning using real coded variable length genetic algorithm for pixel classification. IEEE Trans Geosci Remote Sens 41(5):1075–1081
220. May ACW, Johoson MS (1995) Improved genetic algorithm-based protein structure compaisons: pairwise and multiple superpositions. Protein Eng 8:873–882
221. McGarvey PB, Huang H, Mazumder R, Zhang J, Chen Y, Zhang C, Cammer S, Will R, Odle M, Sobral B, Moore M, Wu CH (2009) Systems integration of biodefense omics data for analysis of pathogen-host interactions and identification of potential targets. PLoS One 4(9):e7162
222. McLachlan GJ, Bean RW, Peel D (2002) A mixture model-based approach to the clustering of microarray expression data. Bioinformatics 18:413–422
223. McLachlan GJ, Do KA, Ambroise C (2004) Analyzing microarray gene expression data. Wiley, Hoboken
224. Medvedovic M, Yeung KY, Bumgarner RE (2004) Bayesian mixture model based clustering of replicated microarray data. Bioinformatics 20(8):1222–1232
225. Mehta M, Agrawal R, Rissanen J (1996) SLIQ: A fast scalable classifier for data mining. In: Proceedings of international conference on extending database technology, Avignon, France
226. Minakuchi Y, Satou K, Konagaya A (2002) Prediction of protein–protein interaction sites using support vector machines. Genome Inform 13:322–323
227. Mitchell TM (1982) Generalization as search. Artif Intell 18(2):203–226
228. Mitra S, De RK, Pal SK (1997) Knowledge-based fuzzy MLP for classification and rule generation. IEEE Trans Neural Netw 8:1338–1350
229. Mitra S, Pal SK (1995) Fuzzy multi-layer perceptron, inferencing and rule generation. IEEE Trans Neural Netw 6:51–63
230. Mitra S, Pal SK (1996) Fuzzy self organization, inferencing and rule generation. IEEE Trans Syst Man Cybern Part A: Syst Humans 26:608–620
231. Moret BME (1982) Decision trees and diagrams. Comput Surv 14(4):593–623
232. Morris GM, Goodsell DS, Halliday RS, Huey R, Hart WE, Belew RK, Olsoni AJ (1998) Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. J Comput Chem 19(14):1639–1662
233. Mukhopadhyay A, Maulik U, Bandyopadhyay S (2009) Multiobjective genetic algorithm-based fuzzy clustering of categorical attributes. IEEE Trans Evol Comput 13(5):991–1005
234. Murthy CA, Bhandari D, Pal SK (1998) ϵ-optimal stopping time for genetic algorithm. Fundam Inform 35(1–4):91–111
235. Mushrif MM, Ray AK (2008) Color image segmentation: rough-set theoretic approach. Patt Recogn Lett 29(4):483–493
236. Narayanan A, Wu XK, Yang ZR (2002) Mining viral protease data to extract cleavage knowledge. Bioinformatics 18:5–13
237. Nasraoui O, Krishnapuram R, Joshi A (1999) Relational clustering based on a new robust estimator with application to web mining. In: Proceedings of the 18th international conference of the North American Fuzzy Information Processing Society, New York, pp 705–709

238. Nasser S, Ranade AR, Sridhart S, Haney L, Korn RL, Gotway MB, Weiss GJ, Kim S (2009) Identifying miRNA and imaging features associated with metastasis of lung cancer to the brain. In: Proceedings of IEEE international conference on bioinformatics and biomedicine, pp 246–251
239. Needleman SB, Wunch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48:443–453
240. Ng RT, Han J (1994) Efficient and effective clustering methods for spatial data mining. In: Proceedings of the 20th international conference on very large databases, Santiago, Chile, pp 144–155
241. Niemann H (1980) Linear and nonlinear mappings of patterns. Patt Recogn 12:83–87
242. Notredame C, Higgins DG (1996) SAGA: sequence alignment by genetic algorithm. Nucl Acids Res 24(8):1515–1524
243. Nowicki R (2008) On combining neuro-fuzzy architectures with the rough set theory to solve classification problems with incomplete data. IEEE Trans Knowl Data Eng 20(9):1239–1253
244. Nowicki R (2009) Rough neuro-fuzzy structures for classification with missing data. IEEE Trans Syst Man Cybern Part B: Cybern 39(6):1334–1347
245. Ono I, Fujiki H, Ootsuka M, Nakashima N, Ono N, Tate S (2002) Global optimization of protein 3-dimensional structures in NMR by a genetic algorithm. Proc Congr Evol Comput 1:303–308
246. Orlowska E (ed) (2010) Incomplete information: rough set analysis. Physica-Verlag, New York
247. Ortega FJ, Moreno-Navarrete JM, Pardo G, Sabater M, Hummel M, Ferrer A, Rodriguez-Hermosa JI, Ruiz B, Ricart W, Peral B, Real JMF (2010) MiRNA expression profile of human subcutaneous adipose and during adipocyte differentiation. PLoS One 5(2):1–9
248. Pal NR, Pal K, Keller JM, Bezdek JC (2005) A possibilistic fuzzy C-means clustering algorithm. IEEE Trans Fuzzy Syst 13(4):517–530
249. Pal SK (2002) Soft computing pattern recognition: principles, integrations and data mining. In: Terano T, Nishida T, Namatame A, Tsumoto S, Ohswa Y, Washio T (eds) Advances in artificial intelligence, vol 2253. Lecture notes in artificial intelligence. Springer, Berlin, pp 261–268
250. Pal SK (2004) Soft data mining, computational theory of perceptions, and rough-fuzzy approach. Inf Sci 163(1–3):5–12
251. Pal SK, Bhandari D (1994) Selection of optimal set of weights in a layered network using genetic algorithms. Inf Sci 80:213–234
252. Pal SK, Ghosh A (1990) Index of area coverage of fuzzy image subsets and object extraction. Patt Recogn Lett 11(12):831–841
253. Pal SK, Ghosh A (1992) Image segmentation using fuzzy correlation. Inf Sci 62(3):223–250
254. Pal SK, Ghosh A (1996) Neuro-fuzzy computing for image processing and pattern recognition. Int J Syst Sci 27(12):1179–1193
255. Pal SK, Ghosh A, Shankar BU (2000) Segmentation of remotely sensed images with fuzzy thresholding, and quantitative evaluation. Int J Remote Sens 21(11):2269–2300
256. Pal SK, Gupta BD, Mitra P (2004) Rough self organizing map. Appl Intell 21(3):289–299
257. Pal SK, King RA, Hashim AA (1983) Automatic gray level thresholding through index of fuzziness and entropy. Patt Recogn Lett 1:141–146
258. Pal SK, Majumder DD (1986) Fuzzy mathemetical approach to pattern recognition. Wiley, Halsted Press, New York
259. Pal SK, Mitra P (2002) Multispectral image segmentation using the rough set-initialized-EM algorithm. IEEE Trans Geosci Remote Sens 40(11):2495–2501
260. Pal SK, Mitra P (2004) Pattern recognition algorithms for data mining. CRC Press, Boca Raton
261. Pal SK, Mitra S (1999) Neuro-fuzzy pattern recognition: methods in soft computing. Wiley, New York
262. Pal SK, Mitra S, Mitra P (2003) Rough-fuzzy MLP: modular evolution, rule generation, and evaluation. IEEE Trans Knowl Data Eng 15(1):14–25

263. Pal SK, Pal A (eds) (2001) Pattern recognition: from classical to modern approaches. World Scientific, Singapore
264. Pal SK, Polkowski L, Skowron A (eds) (2003) Rough-neuro computing: techniques for computing with words. Springer, Heidelberg
265. Pal SK, Skowron A (eds) (1999) Rough-fuzzy hybridization: a new trend in decision making. Springer, Singapore
266. Pal SK, Wang PP (eds) (1996) Genetic algorithms for pattern recognition. CRC Press, Boca Raton
267. Parkinson H, Kapushesky M, Kolesnikov N, Rustici G, Shojatalab M, Abeygunawardena N, Berube H, Dylag M, Emam I, Farne A, Holloway E, Lukk M, Malone J, Mani R, Pilicheva E, Rayner TF, Rezwan F, Sharma A, Williams E, Bradley XZ, Adamusiak T, Brandizi M, Burdett T, Coulson R, Krestyaninova M, Kurnosov P, Maguire E, Neogi SG, Serra PR, Sansone SA, Sklyar N, Zhao M, Sarkans U, Brazma A (2009) ArrayExpress update—from an archive of functional genomics experiments to the atlas of gene expression. Nucl Acids Res 27:D865–D872
268. Patton AL, Punch WF, Goddman ED (1995) A standard GA approach to native protein conformation prediction. In: Proceedings of the 6th international conference on genetic algorithms, pp 574–581
269. Paul S, Maji P (2013) μHEM for identification of differentially expressed miRNAs using hypercuboid equivalence partition matrix. BMC Bioinform 14(1):266
270. Paul S, Maji P (2013) Rough sets for insilico identification of differentially expressed miRNAs. Int J Nanomed 8:63–74
271. Pawlak Z (1991) Rough sets: theoretical aspects of resoning about data. Kluwer, Dordrecht
272. Pearson WR, Lipman DJ (1988) Improved tools for biological sequence comparison. Proc Natl Acad Sci USA 85(8):2444–2448
273. Pedersen JT, Moult J (1997) Protein folding simulations with genetic algorithms and a detailed molecular description. J Mol Biol 269(2):240–259
274. Pereira PM, Marques JP, Soares AR, Carreto L, Santos MAS (2010) MicroRNA expression variability in human cervical tissues. PLoS One 5(7):1–12
275. Pevsner J (2009) Bioinformatics and functional genomics. Wiley-Blackwell, New York
276. Phung SL, Bouzerdoum A (2007) A pyramidal neural network for visual pattern recognition. IEEE Trans Neural Netw 18(2):329–343
277. Polkowski L (2002) Rough sets. Physica-Verlag, Heidelberg
278. Polkowski L, Skowron A (eds) (1998) Rough sets in knowledge discovery, vols 1 and 2. Physica-Verlag, Heidelberg
279. Pudil P, Novovicova J, Kittler J (1994) Floating search methods in feature-selection. Patt Recogn Lett 15:1119–1125
280. Qian N, Sejnowski TJ (1988) Predicting the secondary structure of globular proteins using neural network models. J Mol Biol 202(4):865–884
281. Quackenbush J (2001) Computational analysis of microarray data. Nat Rev Genet 2:418–427
282. Quinlan JR (1993) C4.5: programs for machine learning. Morgan Kaufmann, San Mateo
283. Quteishat A, Lim CP, Tan KS (2010) A modified fuzzy min–max neural network with a genetic-algorithm-based rule extractor for pattern classification. IEEE Trans Syst Man Cybern Part A: Syst Hum 40(3):641–650
284. Rabow AA, Scheraga HA (1996) Improved genetic algorithm for the protein folding problem by use of a cartesian combination operator. Protein Sci 5:1800–1815
285. Raponi M, Dossey L, Jatkoe T, Wu X, Chen G, Fan H, Beer DG (2009) MicroRNA classifiers for predicting prognosis of squamous cell lung cancer. Cancer Res 69(14):5776–5783
286. Reichhardt T (1999) It's sink or swim as a tidal wave of data approaches. Nature 399(6736):517–520
287. Riis SK, Krogh A (1996) Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments. J Comput Biol 3:163–183

288. Ross DT, Scherf U, Eisen MB, Perou CM, Rees C, Spellman P, Iyer V, Jeffrey SS, de RM Van, Waltham M, Pergamenschikov A, Lee JC, Lashkari D, Shalon D, Myers TG, Weinstein JN, Botstein D, Brown PO (2000) Systematic variation in gene expression patterns in human cancer cell lines. Nat Genet 24(3):227–235
289. Roy A, Pal SK (2003) Fuzzy discretization of feature space for a rough set classifier. Patt Recogn Lett 24(6):895–902
290. Russo M (1998) FuGeNeSys: a fuzzy genetic neural system for fuzzy modeling. IEEE Trans Fuzzy Syst 6(3):373–388
291. Saeys Y, Inza I, Larraaga P (2007) A review of feature selection techniques in bioinformatics. Bioinformatics 23(19):2507–2517
292. Safavian SR, Landgrebe D (1991) A survey of decision tree classifier methodology. IEEE Trans Syst Man Cybern 21(3):660–674
293. Saha S, Christensen JP (1994) Genetic design of sparse feedforward neural networks. Inf Sci 79:191–200
294. Salamov A, Solovyev V (1995) Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments. J Mol Biol 247:11–15
295. Salzberg S, Cost S (1992) Predicting protein secondary structure with a nearest-neighbor algorithm. J Mol Biol 227:371–374
296. Sammon JW (1969) A nonlinear mapping for data structure analysis. IEEE Trans Comput 18:401–409
297. Saxena P, Whang I, Voziyanov Y, Harkey C, Argos P, Jayaram M, Dandekar T (1997) Probing flip: a new approach to analyze the structure of a DNA recognizing protein by combining the genetic algorithm, metagenesis and non-canonical DNA target sites. Biochim Biophys Acta—Protein Struct Mol Enzymol 1340(2):187–204
298. Scholkopf B (1997) Support vector learning. Ph.D. thesis, Technische Universilal, Berlin
299. Scholkopf B, Sung KK, Burges CJC, Girosi F, Niyogi P, Poggio T, Vapnik V (1997) Comparing support vector machines with Gaussian kernels to radial basis function classifiers. IEEE Trans Signal Process 45(11):2758–2765
300. Scholkpf B, Smola A, Muller KR (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput 10(5):1299–1319
301. Schrauder MG, Strick R, Schulz-Wendtland R, Strissel PL, Kahmann L, Loehberg CR, Lux MP, Jud SM, Hartmann A, Hein A, Bayer CM, Bani MR, Richter S, Adamietz BR, Wenkel E, Rauh C, Beckmann MW, Fasching PA (2012) Circulating micro-RNAs as potential blood-based markers for early stage breast cancer detection. PLoS One 7(1):1–9
302. Schulze-Kremer S (1995) Molecular bioinformatics: algorithms and applications. Walter de Gruyter, Berlin
303. Schulze-Kremer S (2002) Genetic algorithms and protein folding: methods in molecular biology. Protein Struct Predict: Methods Protoc 143:175–222
304. Searls DB (1996) Sequence alignment through pictures. Trends Genet 12:35–37
305. Searls DB, Murphy KP (1995) Automata-theoretic models of mutation and alignment. In: Proceedings of the 3rd international conference on intelligent systems for molecular biology, The AAAI Press, pp 341–349
306. Segal E, Shapira M, Regev A, Pe'er D, Botstein D, Koller D, Friedman N (2003) Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. Nat Genet 34(2):166–176
307. Seng TL, Bin Khalid M, Yusof R (1999) Tuning of a neuro-fuzzy controller by genetic algorithm. IEEE Trans Syst Man Cybern Part B: Cybern 29(2):226–236
308. Setubal J, Meidanis J (1999) Introduction to computational molecular biology. Thomson, Boston
309. Shafer J, Agrawal R, Mehta M (1996) SPRINT: a scalable parallel classifier for data mining. In: Proceedings of the 22th international conference on very large data bases, Morgan Kaufmann, pp 544–555
310. Shamir R, Sharan R (2000) CLICK: a clustering algorithm for gene expression analysis. In: Proceedings of the 8th international conference on intelligent systems for molecular biology, pp 307–331

311. Shapiro BA, Navetta J (1994) A massively parallel genetic algorithm for RNA secondary structure prediction. J Supercomput 8:195–207
312. Shapiro BA, Wu JC (1996) An annealing mutation operator in the genetic algorithms for RNA folding. Comput Appl Biosci 12:171–180
313. Shapiro BA, Wu JC, Bengali D, Potts MJ (2001) The massively parallel genetic algorithm for RNA folding: MIMD implementation and population variation. Bioinformatics 17(2):137–148
314. Shen Q, Chouchoulas A (1999) Combining rough sets and data-driven fuzzy learning for generation of classification rules. Patt Recogn 32(12):2073–2076
315. Skourikhine A (2000) Phylogenetic tree construction using self-adaptive genetic algorithm. In: Proceedings of the IEEE international symposium on bioinformatics and biomedical engineering, pp 129–134
316. Skowron A, Swiniarski R (2001) Rough sets in pattern recognition. In: Pal SK, Pal A (eds) Pattern recognition: from classical to modern approaches. World Scientific, Singapore, pp 385–428
317. Slezak D (1996) Approximate reducts in decision tables. In: Proceedings of the 6th international conference on information processing and management of uncertainty in knowledge-based systems, pp 1159–1164
318. Slezak D (2007) Rough sets and few-objects-many-attributes problem: the case study of analysis of gene expression data sets. In: Proceedings of the frontiers in the convergence of bioscience and information technologies, pp 233–240
319. Slezak D, Betlinski P (2012) A role of (not) crisp discernibility in rough set approach to numeric feature selection. In: Hassanien AE, Salem ABM, Ramadan R, Kim TH (eds) Advanced machine learning technologies and applications, vol 322, pp 13–23
320. Slezak D, Janusz A (2011) Ensembles of bireducts: towards robust classification and simple representation. In: Kim TH, Adeli H, Slezak D, Sandnes F, Song X, Chung KI, Arnett KP (eds) Future generation information technology, vol 7105, pp 64–77
321. Slezak D, Widz S (2010) Evolutionary inspired optimization of feature subset ensembles. In: Proceedings of the 2nd world congress on nature and biologically inspired computing, pp 437–442
322. Slezak D, Widz S (2010) Is it important which rough-set-based classifier extraction and voting criteria are applied together? In: Proceedings of the 7th international conference on rough sets and current trends in computing, pp 187–196
323. Slezak D, Widz S (2011) Rough-set-inspired feature subset selection, classifier construction, and rule aggregation. In: Yao JT, Ramanna S, Wang G, Suraj Z (eds) Rough sets and knowledge technology, vol 6954, pp 81–88
324. Slezak D, Wroblewski J (2007) Roughfication of numeric decision tables: the case study of gene expression data. In: Proceedings of the 2nd international conference on rough sets and knowledge technology. Springer, Berlin, pp 316–323
325. Smith TF, Waterman MS (1981) Identification of common molecular sequences. J Mol Biol 147:195–197
326. Somorjai RL, Dolenko B, Baumgartner R (2003) Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions. Bioinformatics 19(12):1484–1491
327. Stosic BD, Fittipaldi IP (1997) Pattern recognition via ISING model with long range interactions. Phys A: Stat Mech Appl 242(3–4):323–331
328. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, Mesirov JP (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci USA 102(43):15545–15550
329. Sun Z, Xia X, Guo Q, Xu D (1999) Protein structure prediction in a 210-type lattice model: parameter optimization in the genetic algorithm using orthogonal array. J Protein Chem 18(1):39–46

330. Sun ZL, Au KF, Choi TM (2007) A neuro-fuzzy inference system through integration of fuzzy logic and extreme learning machines. IEEE Trans Syst Man Cybern Part B: Cybern 37(5):1321–1331
331. Suna F, Zhanga W, Xionga G, Yanb M, Qian Q, Lia J, Wanga Y (2010) Identification and functional analysis of the MOC1 interacting protein 1. J Genet Genomics 37(1):69–77
332. Sventik V, Wang T, Tong C, Liaw A, Sheridan RP, Song Q (2005) Boosting: an ensemble learning tool for compound classification and QSAR modeling. J Chem Inf Model 45(3):786–799
333. Swain P, Hauska H (1977) The decision tree classifier design and potential. IEEE Trans Geosci Electron 15:142–147
334. Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander ES, Golub TR (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. Proc Natl Acad Sci USA 96(6):2907–2912
335. Tavazoie S, Hughes D, Campbell MJ, Cho RJ, Church GM (1999) Systematic determination of genetic network architecture. Nat Genet 22(3):281–285
336. Theodoridis S, Koutroumbas K (2008) Pattern recognition. Elsevier Science, New York
337. Thompson K (1968) Regular expression search algorithm. Commun ACM 11(6):419–422
338. Thomson R, Hodgman C, Yang ZR, Doyle AK (2003) Characterising proteolytic cleavage site activity using bio-basis function neural network. Bioinformatics 19(14):1741–1747
339. Tieri P, Fuente A, Termanini A, Franceschi C (2011) Integrating omics data for signaling pathways, interactome reconstruction, and functional analysis. In: Mayer B (ed) Bioinformatics for omics data, methods in molecular biology, vol 719. Springer, New York, pp 415–433
340. Timothy TS, Diego DD, David D, James JJ (2003) Inferring genetic networks and identifying compound mode of action via expression profiling. Science 301(5629):102–105
341. Tino P, Zhao H, Yan H (2011) Searching for coexpressed genes in three-color cDNA microarray data using a probabilistic model-based Hough transform. IEEE/ACM Trans Comput Biol Bioinform 8(4):1093–1107
342. Tobias OJ, Seara R (2002) Image segmentation by histogram thresholding using fuzzy sets. IEEE Trans Image Process 11(12):1457–1465
343. Tou JT, Gonzalez RC (1974) Pattern recognition principles. Addison-Wesley, Reading
344. Tung AKH, Han J, Lakshmanan LVS, Ng RT (2001) Constraint-based clustering in large databases. In: Proceedings of the international conference on database theory,. London, UK, pp 405–419
345. Tung AKH, Hou J, Han J (2001) Spatial clustering in the presence of obstacles. In: Proceedings of the international conference on data engineering, Heidelberg, Germany, pp 359–367
346. Turner AP, Lones MA, Fuente LA, Stepney S, Caves LSD, Tyrrell AM (2013) The incorporation of epigenetics in artificial gene regulatory networks. Biosystems 112(2):56–62
347. Tzur G, Israel A, Levy A, Benjamin H, Meiri E, Shufaro Y, Meir K, Khvalevsky E, Spector Y, Rojansky N, Bentwich Z, Reubinoff BE, Galun E (2009) Comprehensive gene and microRNA expression profiling reveals a role for microRNAs in human liver development. PLoS One 4(10):1–13
348. Uberbacher E, Mural R (1991) Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach. Proc Natl Acad Sci USA 88(24):11261–11265
349. Unger R, Moult J (1993) On the applicability of genetic algorithms to protein folding. Proc Hawaii Int Conf Syst Sci 1:715–725
350. Valdes JJ, Barton AJ (2005) Relevant attribute discovery in high dimensional data: application to breast cancer gene expressions. In: Proceedings of the 1st international conference on rough sets and knowledge technology, Springer-Berlin, Heidelberg, pp 482–489
351. Vapnik V (1995) The nature of statistical learning theory. Springer, New York
352. Vignes M, Forbes F (2009) Gene clustering via integrated Markov models combining individual and pairwise features. IEEE/ACM Trans Comput Biol Bioinform 6(2):260–270
353. Vizan P, Sanchez-Tena S, Alcarraz-Vizan G, Soler M, Messeguer R, Pujol M, Lee WNP, Cascante M (2009) Characterization of the metabolic changes underlying growth factor angiogenic activation: identification of new potential therapeutic targets. Carcinogenesis 30(6):946–952

354. Wang C, Yang S, Sun G, Tang X, Lu S, Neyrolles O, Gao Q (2011) Comparative miRNA expression profiles in individuals with latent and active tuberculosis. PLoS One 6(10):1–11
355. Wang H, Wang Z, Li X, Gong B, Feng L, Zhou Y (2011) A robust approach based on Weibull distribution for clustering gene expression data. Algorithms Mol Biol 6(1):14
356. Wang J, Chen B, Wang Y, Wang N, Garbey M, Tran-Son-Tay R, Berceli SA, Wu R (2013) Reconstructing regulatory networks from the dynamic plasticity of gene expression by mutual information. Nucl Acids Res 41(8):e97
357. Wang L, Zhu J, Zou H (2008) Hybrid huberized support vector machines for microarray classification and gene selection. Bioinformatics 24(3):412–419
358. Wang WY, Li YH (2003) Evolutionary learning of BMF fuzzy-neural networks using a reduced-form genetic algorithm. IEEE Trans Syst Man Cybern Part B: Cybern 33(6):966–976
359. Wang Y, Dunham MH, Waddle JA, Mcgee M (2006) Classifier fusion for poorly-differentiated tumor classification using both messenger RNA and micro RNA expression profiles. In: Proceedings of the international conference on computational systems, bioinformatics, pp 1–5
360. Waterman M (1990) RNA structure prediction. Methods in enzymology. Academic, San Diego, p 164
361. Wei Q, Chen G (1999) Mining generalized association rules with fuzzy taxonomic structures. In: Proceedings of the 18th international conference of the North American Fuzzy Information Processing Society, New York, pp 477–481
362. Whitley D, Starkweather T, Bogart C (1990) Genetic algorithms and neural networks: optimizing connections and connectivity. Parallel Comput 14:347–361
363. Widz S, Revett K, Slezak D (2005) A hybrid approach to MR imaging segmentation using unsupervised clustering and approximate reducts. In: Proceedings of the 10th international conference on rough sets, fuzzy sets, data mining, and granular, computing, pp 372–382
364. Widz S, Revett K, Slezak D (2005) A rough set-based magnetic resonance imaging partial volume detection system. In: Proceedings of the 1st international conference on pattern recognition and machine intelligence, pp 756–761
365. Widz S, Slezak D (2007) Approximation degrees in decision reduct-based MRI segmentation. In: Proceedings of the frontiers in the convergence of bioscience and information technologies, pp 431–436
366. Wiese KC, Glen E (2003) A permutation-based genetic algorithm for the RNA folding problem: a critical look at selection strategies, crossover operators, and representation issues. Biosystems 72(1–2):29–41
367. Winston PH (1975) Learning structural descriptions from examples. In: Winston PH (ed) The psycology of computer vision. McGraw Hill, New York
368. Wong WC, Cho SY, Quek C (2009) R-POPTVR: a novel reinforcement-based POPTVR fuzzy neural network for pattern classification. IEEE Trans Neural Netw 20(11):1740–1755
369. Wroblewski J (1995) Finding minimal reducts using genetic algorithms. In: Proceedings of the 2nd annual joint conference on information sciences, pp 186–189
370. Xiao K, Ho SH, Hassanien AE (2008) Automatic unsupervised segmentation methods for MRI based on modified fuzzy C-means. Fundam Inform 87(3–4):465–481
371. Xiaodong C, Giannakis GB (2006) Identifying differentially expressed genes in microarray experiments with model-based variance estimation. IEEE Trans Signal Process 54(6):2418–2426
372. Xing EP, Karp RM (2001) CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. Bioinformatics 17(1):306–315
373. Xu Y, Olman V, Xu D (2002) Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees. Bioinformatics 18(4):536–545
374. Yang JM, Kao CY (2000) A family competition evolutionary algorithm for automated docking of flexible ligands to proteins. IEEE Trans Inform Technol Biomed 4(3):225–237
375. Yang ZR (2004) Biological application of support vector machines. Brief Bioinform 5(4):328–338

376. Yang ZR (2005) Prediction of caspase cleavage sites using Bayesian bio-basis function neural networks. Bioinformatics 21(9):1831–1837
377. Yang ZR (2010) Machine learning approaches to bioinformatics. World Scientific Publishing Company, Hackensack
378. Yang ZR, Chou KC (2004) Predicting the O-linkage sites in glycoproteins using bio-basis function neural networks. Bioinformatics 20(6):903–908
379. Yang ZR, Thomson R (2005) Bio-basis function neural network for prediction of protease cleavage sites in proteins. IEEE Trans Neural Netw 16(1):263–274
380. Yang ZR, Thomson R, McNeil P, Esnouf R (2005) RONN: use of the bio-basis function neural network technique for the detection of natively disordered regions in proteins. Bioinformatics 21(16):3369–3376
381. Yanhua C, Ming D, Rege M (2007) Gene expression clustering: a novel graph partitioning approach. In: Proceedings of international joint conference on neural networks, pp 1542–1547
382. Yeung K, Medvedovic M, Bumgarner R (2003) Clustering gene-expression data with repeated measurements. Genome Biol 4(5):R34
383. Yeung KY, Fraley C, Murua A, Raftery AE, Ruzz WL (2001) Model-based clustering and data transformations for gene expression data. Bioinformatics 17(10):977–987
384. Yu Z, Wong HS, Wang H (2007) Graph-based consensus clustering for class discovery from gene expression data. Bioinformatics 23(21):2888–2896
385. Zadeh LA (1965) Fuzzy sets. Inform Control 8:338–353
386. Zadeh LA (1979) Fuzzy sets and information granularity. In: Gupta M, Ragade R, Yager R (eds) Advances in fuzzy set theory and applications. North-Holland Publishing Co., Amsterdam, pp 3–18
387. Zadeh LA (1994) Fuzzy logic, neural networks, and soft computing. Commun ACM 37:77–84
388. Zhang J (2005) Modeling and optimal control of batch processes using recurrent neuro-fuzzy networks. IEEE Trans Fuzzy Syst 13(4):417–427
389. Zhang T, Ramakrishnan R, Livny M (1996) BIRCH: an efficient data clustering method for very large databases. In: Proceedings of the ACM-SIGMOD international conference on management of data, Montreal, Canada, pp 103–114
390. Zhang X, Zhao XM, He K, Lu L, Cao Y, Liu J, Hao JK, Liu ZP, Chen L (2012) Inferring gene regulatory networks from gene expression data by PC-algorithm based on conditional mutual information. Bioinformatics 28:98–104
391. Zhang Y, Rajapakse JC (eds) (2008) Machine learning in bioinformatics. Wiley, New York
392. Zhao H, Shen J, Medico L, Wang D, Ambrosone CB, Liu S (2010) A pilot study of circulating miRNAs as potential biomarkers of early stage breast cancer. PLoS One 5(10):1–12
393. Zheng M, Wu JN, Huang YX, Liu GX, Zhou Y, Zhou CG (2012) Inferring gene regulatory networks by singular value decomposition and gravitation field algorithm. PLoS One 7(12):e51141
394. Zhou L, Zenebe A (2008) Representation and reasoning under uncertainty in deception detection: a neuro-fuzzy approach. IEEE Trans Fuzzy Syst 16(2):442–454
395. Zuker M, Striegler P (1981) Optimal computer folding of large RNA secondary sequences using thermodynamics and auxiliary information. Nucl Acids Res 9:133–148

Part I
Classification

Chapter 2
Neural Network Tree for Identification of Splice Junction and Protein Coding Region in DNA

2.1 Introduction

The internetworked society has been experiencing an explosion of biological data. However, the explosion is paradoxically acting as an impediment in acquiring knowledge. The meaningful interpretation of this large volume of biological data is increasingly becoming difficult. Consequently, researchers, practitioners, and entrepreneurs from diverse fields are trying to develop sophisticated techniques to store, analyze, and interpret this biological data for knowledge extraction, which leads to evolve the new field called bioinformatics. This field has arised in parallel with the development of automated high-throughput methods of pattern recognition and machine learning. The development of high-throughput methods for biological and biochemical discovery yields a variety of experimental data such as DNA sequences, gene expression patterns, chemical structures, and so forth. Hence, bioinformatics encompasses everything from data storage and retrieval to the identification and presentation of features within data such as finding genes within DNA sequence, finding similarities between sequences, structural predictions, and correlation between sequence variation and clinical data [1, 4–6, 25].

Two of the important problems in bioinformatics are splice-site or splice-junction prediction and identification of protein coding regions in DNA sequences. Genes contain their information as a specific sequence of nucleotides or bases that are found in DNA molecules. These specific sequences of bases encode instructions on how to make proteins. The regions of a gene that code for proteins are termed as exons . These exons occupy only a small region of the gene. Whereas in prokaryotic gene, the mRNA (messenger ribonucleic acid) is a mere transcribed copy of the DNA, in eukaryotic gene, the RNA copy of the DNA contains noncoding segments, which are termed as introns, and they should be precisely spliced out to produce the mRNA. This means that introns are parts of a gene that are not used in protein synthesis and exons are the protein coding regions in that gene. The points at which DNA is removed are known as splice sites. The splice-site identification problem is to determine into which of the following three categories a specified location in a

P. Maji and S. Paul, *Scalable Pattern Recognition Algorithms*,
DOI: 10.1007/978-3-319-05630-2_2,

DNA sequence falls: (1) exon/intron borders, referred to as donors; (2) intron/exon borders, referred to as acceptors; and (3) neither. Another important problem is the identification of protein coding region, that is, exon in anonymous sequences of DNA. Identifying the coding regions and splice sites is of vital importance in understanding the processing of genes.

Many new mathematical or computational approaches are being introduced as well as existing ones getting refined, but the search for new and better solutions continues specifically to analyze large volume of DNA data sets generated in the internetworked society of cyber-age. To address these problems, a new neural network tree (NNTree) based pattern classifier [11, 12] is presented in this chapter for finding the splice-site and protein coding regions in DNA sequences. The idea of this new method is to use the framework of decision tree and neural network.

Over the years, the decision trees (DT) are successfully used in many diverse areas such as radar signal classification, character recognition, remote sensing, medical diagnosis, expert system, speech recognition, and also in other different fields [18]. The decision tree classifier is one of the possible approaches in multistage decision making. The most important feature of DT is their capability to break up a complex decision into a union of several simpler decisions hoping the final solution obtained this way would resemble the intended desired solution. On the other hand, the subject of artificial neural network (ANN) has become very popular in many areas such as signal processing and pattern recognition [8, 10, 23, 26]. Additionally, neural networks are models of nonsymbolic approaches. However, nonsymbolic learners are usually black boxes. It is not known what has been learned ever if correct answers are got. Another key problem in using neural networks is that the number of free parameters is usually too large to be determined efficiently.

Even though neural networks and DT are two very different techniques for pattern recognition or classification, both are capable of generating arbitrarily complex decision boundaries from a given set of training samples or training examples, and usually neither has to make any assumptions about the distribution of the underlying processes. The neural networks are usually more capable of providing incremental learning than DT, whereas decision trees are sequential in nature, compared to massive parallelism usually present in neural networks. Thus, DT are typical models for symbolic approaches, and neural networks are models for nonsymbolic approaches. Basically, symbolic approaches can provide comprehensive rules but cannot adapt to changing environments efficiently. On the contrary, nonsymbolic approaches can adapt to changing environments but cannot provide comprehensible rules.

In this background, many pattern classifiers have been proposed, integrating the advantages of decision tree and neural network. One of the early pattern classifiers based on this concept is Entropy Nets due to Sethi [20]. It derives a multilayer feedforward neural network from a decision tree. The knowledge represented by the decision tree is translated into the architecture of a neural network whose connections can be retrained by a back propagation algorithm. On the other hand, the ANN-DT [19] uses neural network to generate outputs for examples interpolated from the training data set and then extracts a univariate binary decision tree from the network. Another method which also extracts decision tree from neural network is reported

in [24]. The design of a tree-structured neural network using genetic programming is proposed in [9]. In [7, 22, 27–29], designs of NNTrees have been introduced. The NNTree is a decision tree with each nonterminal node being a neural network. In [21], Sethi and Yoo have proposed a decision tree whose hierarchy of splits is determined in a global fashion by a neural learning algorithm. Recently, Zhou and Chen [30] have introduced a hybrid learning approach named HDT that embeds neural network in some leaf nodes of a binary decision tree.

To design an NNTree, the most important and time-consuming step is splitting the nonterminal nodes of the tree. There are many criteria for splitting nonterminal nodes. One of the most popular criteria is the information gain ratio which is used in C4.5 [18]. Designs of NNTree have so far mostly concentrated around binary tree with information gain ratio used to partition the available data set at each nonterminal node [7, 22, 27–29]. However, this structure generates larger height of the tree. In effect, it increases classification time and error rate in classifying test samples. Also none of the work has so far dealt with the application of the NNTree to biological data set.

In the above background, this chapter presents the design and applications of an NN-based tree-structured pattern classifier (NNTree) to address the problem of finding splice-site and protein coding region in DNA sequences [11, 12]. The NNTree reported here adopts an approach which is completely different from the methods mentioned in [7, 22, 24, 27–30]. The neural networks used in this design are multilayer perceptrons (MLP) with m output nodes; m being the number of classes in the given data set. Unlike [7, 22, 27–30], the NNTree designed here splits each nonterminal node by maximizing (respectively, minimizing) classification accuracy (respectively, classification error) of the MLP rather than using information gain ratio. So, the current design always generates a reduced height m-ary tree. The backpropagation algorithm is used recursively at each nonterminal node to find a multilayer perceptron. The effectiveness of the new algorithm, along with a comparison with individual components of the hybrid scheme as well as other related algorithms, has been demonstrated on several benchmark data sets.

The structure of the rest of this chapter is as follows: Sect. 2.2 presents the design of a neural network based tree-structured pattern classifier, called NNTree. Sections 2.3 and 2.4 present the application of the NNTree in splice-junction and protein coding region identification problems, respectively. In order to validate the design of current model, extensive experimental results are also reported in these sections. Concluding remarks are given in Sect. 2.5.

2.2 Neural Network Based Tree-Structured Pattern Classifier

A neural network tree (NNTree) is a decision tree with each intermediate or nonterminal node being a MLP. It is constructed by partitioning the training set consisting of feature vectors and their corresponding class labels in such a way as to recursively generate the tree. This procedure involves three steps: splitting nodes, determining

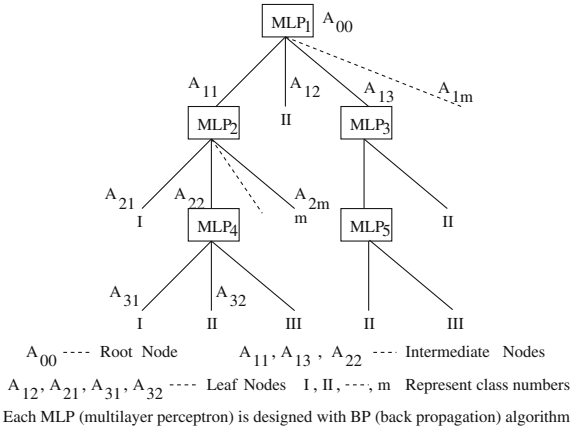

Fig. 2.1 MLP-based tree-structured pattern classifier

which nodes are terminal nodes, and assigning class labels to terminal nodes. In this tree, a leaf or terminal node covers the set or subset of elements of only one class. By contrast, an intermediate node covers the set or subset of elements belonging to more than one class. Thus, the NNTrees are class discriminators which recursively partition the training set to get nodes belonging to a single class. Figure 2.1 shows an MLP-based NNTree.

To classify a training set $S = \{S_1, \ldots, S_i, \ldots, S_m\}$ consisting of m classes, an MLP has to be designed with m neurons in output layer. If a pattern belongs to ith class, ith output neuron is selected. That is, the content of the ith neuron is 1. At this moment, the content of all other output neurons are 0s. So, the output layer represents m distinct m-dimensional vectors, each representing a unique location or node. Thus, the training set S gets distributed into m locations or nodes using an MLP.

Let, $\acute{S}$ be the set of elements in a node. If $\acute{S}$ belongs to only one class, then label that node as that class. Otherwise, this process is repeated recursively for each node until all the patterns in each node or location belong to only one class. Single or multiple nodes of the tree may form a leaf or terminal node representing a class, or it may be an intermediate or nonterminal node. A leaf node represents a location that contains the set or subset of elements of only one class. By contrast, an intermediate node refers to a location that contains the elements belonging to more than one class. In effect, an intermediate node represents the decision to build a new MLP for the elements of multiple classes covered by the location of the earlier level. The above discussions are formalized next.

Input: Training set $S = \{S_1, \ldots, S_i, \ldots, S_m\}$
Output: NNTree (set of MLPs)

Partition(S, m);

Partition$(\tilde{S}, \tilde{m})$

1. Generate an MLP with $\tilde{m}$ output neurons.
2. Distribute the training set $\tilde{S}$ into $\tilde{m}$ locations (nodes).
3. Evaluate the distribution of patterns in each node.
4. If all the patterns ($\acute{S}$) of a location (node) belong to one particular class, then label the location (leaf node) for that class.
5. If for the set of patterns ($\acute{S}$) of a location belonging to $\acute{m}$ classes, **Partition**($\acute{S}$, $\acute{m}$).
6. End.

In Fig. 2.1, the node A_{00} is the root node. So, the MLP_1 corresponding to node A_{00} distributes the training set $S = \{S_1, \ldots, S_i, \ldots, S_m\}$ into m locations denoted by $A_{11}, A_{12}, \ldots, A_{1m}$. Now, A_{11} is an intermediate node as the elements covered by this location belong to multiple classes (here m) which are distributed again by MLP_2 into m number of locations - $A_{21}, A_{22}, \ldots, A_{2m}$. A_{13} is also an intermediate node, but it covers the elements of classes II and III only. So, MLP_3 corresponding to node A_{13} generates two locations or nodes to distribute these elements. Whereas A_{12} is a leaf or terminal node as it contains the elements of only one class (here class II). Similarly, $A_{21}, A_{2m}, A_{31}, A_{32}, \ldots,$ are the leaf or terminal nodes as they cover the elements of single class.

In designing an NNTree for a given data set, there are two options:

1. design an NNTree that correctly classifies all the training samples (referred to as a perfect tree), and select the smallest perfect tree; and
2. construct an NNTree that is not perfect but has the smallest possible error rate in classification of test samples.

The second type of tree is of greater interest for real life pattern recognition task. Regardless of the type of tree (perfect or otherwise), it is usually desirable to keep the size of the tree as small as possible. Because, smaller trees are more efficient both in terms of tree storage requirements and test time; and tend to generalize better for the unseen test samples that are less sensitive to the statistical irregularities and idiosyncrasies of the training data. So, the basic criteria for the NNTree design are as follows:

1. minimize error rate that would lead to maximum classification accuracy;
2. less number of nodes in the tree, that is, minimum number of locations of the selected NNTree; and
3. least height of the NNTree.

2.2.1 Selection of Multilayer Perceptron

Following two steps are implemented at each intermediate node to pick up the best possible NNTree:

1. evaluation of candidate MLPs, that is, evaluation of distribution of the elements of different classes in different locations of an MLP; and

2. selection of a location using the best distribution in the intermediate nodes ensuring maximum classification accuracy.

The complexity lies in determining the best distribution for each intermediate node. The optimal NNTree is evolved through the application of back propagation algorithm [8, 10] recursively at each intermediate node.

2.2.2 Splitting and Stopping Criteria

Splitting an intermediate node involves the design of a new MLP to classify the subset of input elements of different classes covered by the node or location of the MLP of earlier level of the tree. A location is considered as a leaf node if all the training examples falling into the current location belong to the same class. In other words, a node (location) is split as long as there are class elements that belong to different classes.

To avoid overfitting, a prepruning strategy is needed. Let, C_{ij} be the number of elements of class j covered by ith location, where $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, m$; and β_i indicates the uniformity of the distribution of class elements in the ith location. The value of β_i corresponding to ith node (location) is given by

$$\beta_i = \frac{\mathscr{A}}{\mathscr{B}}; \tag{2.1}$$

$$\text{where } \mathscr{A} = \max_j\{C_{ij}\}; \quad \text{and } \mathscr{B} = \sum_{j=1}^{m} C_{ij}. \tag{2.2}$$

The diversity of the current node is measured as

$$\delta_i = 1 - \beta_i = 1 - \frac{\mathscr{A}}{\mathscr{B}}. \tag{2.3}$$

When current node (location) is to split, its β_i value is measured and compared with a threshold value ε ($= 0.9988$).

1. If $\beta_i < \varepsilon$, then current node is split. That is, partition the examples of ith location.
2. If $\beta_i > \varepsilon$, that is, $\delta_i \simeq 0$, then the learning process terminates and the ith location indicates the class j for which C_{ij} is maximum. The future class elements falling into current node (location) are classified to the most probable class of current node, that is, the class that has the maximum number of training examples in current location.
3. In some cases, even $\beta_i < \varepsilon$, there exists a possibility where desired MLP is not available. That is, it is not possible to find an MLP which can distribute the given training examples into multiple locations. This occurs when the training examples of different classes are highly correlated. In that case, the learning process is

terminated. The future class elements are classified as class j for which C_{ij} is maximum, that is, the most probable class of the current node.

Suppose, after evaluating the distribution of patterns of each class, the pattern set S is partitioned into S_a and S_b, where S_a and S_b represent the pattern set belonging to leaf nodes and intermediate nodes, respectively. The goodness of the splitting or partition is given by the figure of merit (FM), where

$$\text{FM} = \frac{\mid S_a \mid}{\mid S \mid} = 1 - \frac{\mid S_b \mid}{\mid S \mid}; \tag{2.4}$$

where $S_a \cup S_b = S$ and $\mid S \mid$ represent the cardinality of the set S. The value of FM indicates the classification accuracy of an intermediate or nonterminal node.

For ease of discussions, in the rest of the chapter, following terminologies are used:

- η and α represent the learning rate and momentum constant of back propagation algorithm.
- H_n is the number of neurons in the hidden layer of MLP.
- L is the depth of the NNTree, which is equal to the number of levels from the root to the leaf nodes.
- B represents the breadth of the NNTree, which is the number of intermediate nodes in each level of the tree.
- Classification accuracy is defined as the percentage of samples that are correctly classified.
- Classification time is defined as the time required to classify all the samples.

The NNTree is implemented in C language and run in LINUX environment having machine configuration Pentium IV, 3.2 GHz, 1 MB cache, and 1 GB RAM.

2.3 Identification of Splice-Junction in DNA Sequence

In this section, the application of the NNTree in finding the splice junction in anonymous sequences of DNA is presented. The performance of the NNTree is evaluated for benchmark data set analyzing classification accuracy.

In bioinformatics, one of the major tasks is the recognition of certain DNA subsequences those are important in the expression of genes. Basically, a DNA sequence is a string over alphabet $\mathbf{D} = \{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{T}\}$. DNA contains the information by which a cell constructs protein molecules. The cellular expression of protein proceeds by the creation of a messenger ribonucleic acid (mRNA) copy from the DNA template. This mRNA is then translated into a protein. One of the most unexpected findings in molecular biology is that large pieces of the mRNA are removed before it is translated further [2]. The utilized sequences are known as exons while the removed sequences are known as introns, or intervening sequences. The points at which DNA is removed are known as splice junctions. The splice-junction problem is to determine into which of the following three categories a specified location in a DNA sequence falls: (1)

Table 2.1 Classification accuracy for $\eta = 0.50$ and $\alpha = 0.70$

Depth of Tree	$H_n = 10$			$H_n = 15$		
	Training	Testing	Breadth	Training	Testing	Breadth
1	99.2	91.4	1	98.7	91.6	1
2	99.6	93.5	3	99.7	94.2	2
3	99.9	93.5	1	99.9	94.2	2

Table 2.2 Classification accuracy for $\alpha = 0.70$

Depth of Tree	$\eta = 0.90$ and $H_n = 10$			$\eta = 0.80$ and $H_n = 15$		
	Training	Testing	Breadth	Training	Testing	Breadth
1	84.2	81.6	1	82.3	82.6	1
2	89.3	84.3	3	87.6	84.9	3
3	91.7	84.6	6	90.3	85.6	6
4	93.7	84.8	8	93.5	85.6	8
5	95.0	84.9	5	95.3	85.7	8
6	96.4	85.1	6	96.5	85.7	7

exon/intron borders, referred to as donors; (2) intron/exon borders, referred to as acceptors; and (3) neither.

2.3.1 Description of Data Set

The data set used in this problem is a processed version of the Irvine Primate splice-junction database [14]. Each of the 3,186 examples in the database consists of a window of 60 nucleotides, each represented by one of four symbolic values ({**A**, **C**, **G**, **T**}) and the classification of the middle point in the window as one of intron–exon boundary, or neither of these. Processing involved the removal of four examples, conversion of the original 60 symbolic attributes to 180 binary attributes and the conversion of symbolic class labels to numeric labels. The training set of 2,000 is chosen randomly from the data set and the remaining 1,186 examples are used as the test set.

2.3.2 Experimental Results

The experimental results on data set reported in earlier subsection are presented in Tables 2.1, 2.2, 2.3, 2.4, Figs. 2.2, 2.3. Tables 2.1 and 2.2 represent the classification accuracy of both training and test samples for different values of η, α, and H_n. The classification accuracy of training and testing confirms that the

Table 2.3 Performance of NNTree and C4.5 on splice-junction database

Algorithms/ methods	Classification accuracy (%)	Total nodes	Height of tree	Classification time (ms)
NNTree	94.2	5	3	517
C4.5	93.3	127	12	655

Table 2.4 Classification accuracy for splice-junction database

Algorithms	Accuracy (%)
NNTree	94.2
MLP	91.4
Bayesian	90.3
C4.5	93.3
CA	87.9

NNTree can generalize the splice-junction database irrespective of the values of η, α, and H_n. From Figs. 2.2 and 2.3, it is seen that the standard deviations of both training and testing accuracy reduce with the increase in L.

For splice-junction database, at $\eta = 0.50$ and $\alpha = 0.70$, most of the nodes of the NNTree have β_i values greater than ε. So, the learning process terminates at $L = 3$ irrespective of the value of H_n. Whereas, for other values of η and α, the values of β_i for most of the nodes of the NNTree are less than ε when $L \leq 6$. So, for $L \leq 6$, most of the nodes are intermediate nodes. At $L = 7$, though $\beta_i < \varepsilon$ for most of the nodes, the training examples of different classes are so correlated that an MLP cannot be found corresponding to each node, which can classify the data set present at that node. Hence, the NNTree stops to grow.

Table 2.3 compares the performance of the NNTree with C4.5, a popular decision tree algorithm [18], with respect to classification accuracy, total number of intermediate nodes, height of the tree, and classification time. For splice-junction database, the classification accuracy of the NNTree is higher than that of the C4.5, while the number of intermediate nodes, height of the tree, and classification time of the NNTree are significantly smaller than C4.5.

Finally, Table 2.4 compares the classification accuracy of the NNTree with that of different classification algorithms, namely, Bayesian [3], C4.5 [18], MLP [8, 10], and cellular automata (CA) [13]. The experimental results of Table 2.4 clearly establish the fact that the classification accuracy of the NNTree is higher than that of several other classification algorithms.

2.4 Identification of Protein Coding Region in DNA Sequence

This section presents the application of the NNTree for finding protein coding (exon) regions in anonymous sequences of DNA. The performance of the NNTree is evaluated for few sequences and an analysis regarding the accuracy of the method is also presented.

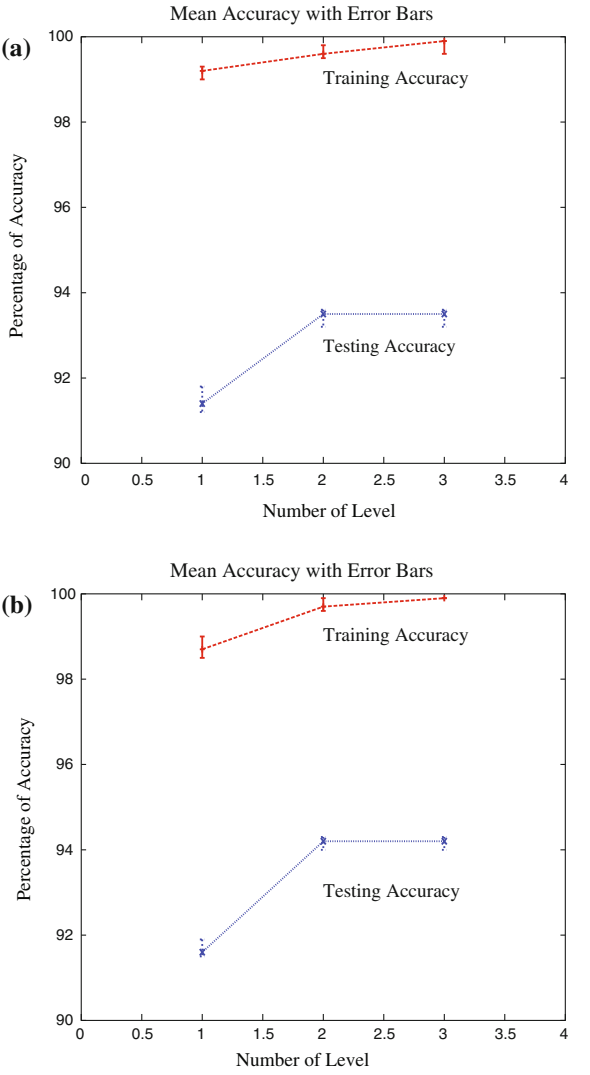

Fig. 2.2 Performance of NNTree on splice-junction database for $\eta = 0.50$ and $\alpha = 0.70$. **a** $H_n = 10$; **b** $H_n = 15$

Over the past 20 years, researchers have identified a number of features of exonic DNA that appear to be useful in distinguishing between coding and noncoding regions [1, 4, 5, 25]. These features include both statistical and information-theoretic measures, and in many cases are based on knowledge of the biology underlying DNA sequences and transcription processes. These features are summarized in a survey by Fickett and Tung [6], who also have developed several benchmark features and databases for future experiments on this problem.

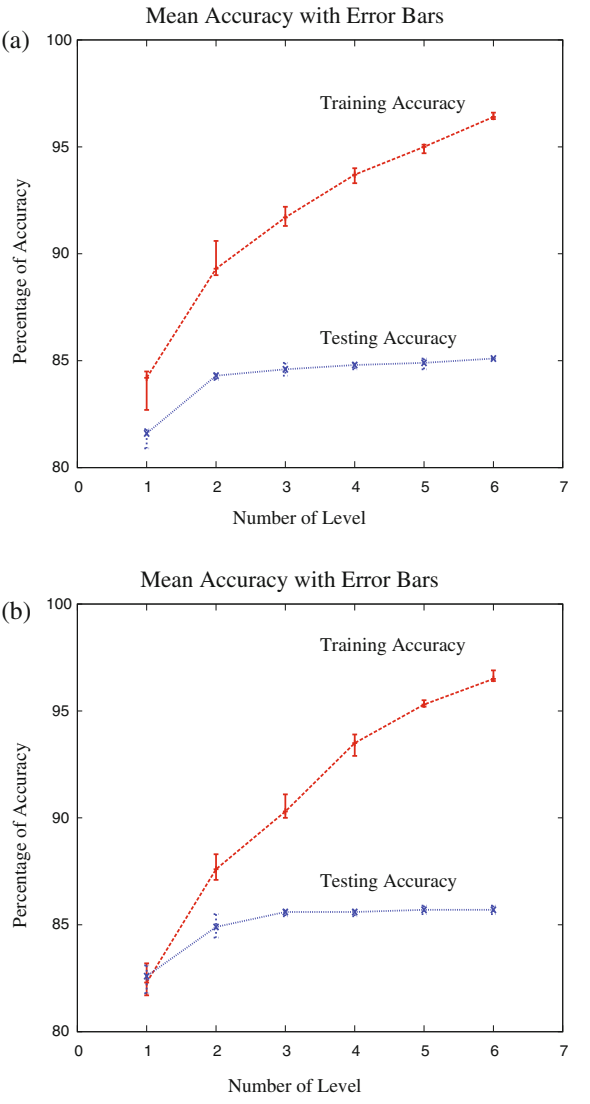

Fig. 2.3 Performance of the NNTree on splice-junction database for $\alpha = 0.70$. **a** $\eta = 0.90$ and $H_n = 10$; **b** $\eta = 0.80$ and $H_n = 15$

Previous research on automatic identification of protein coding regions has considered methods such as linear discriminants [5, 6] and neural networks [4, 25]. These systems have used measures such as codon frequencies, dicodon frequencies,

Table 2.5 Benchmark data sets proposed by Fickett and Tung

Data Set	Human 54	Human 108	Human 162
Training set—coding	20,456	7,086	3,512
Training set—noncoding	125,132	58,118	36,502
Training set total	145,588	65,204	40,014
Test set—coding	22,902	8,192	4,226
Test set—noncoding	122,138	57,032	35,602
Test set total	145,040	65,224	39,868

fractal dimensions, repetitive hexamers, and other features to identify exons in relatively short DNA sequences. The standard experimental study considers a limited window (that is, a subsequence) of a fixed length, for example 100 base pairs, and computes features based on that window alone. The goal is to identify the window as either all-coding or all-noncoding.

2.4.1 Data and Method

The data used for this study are the human DNA data collected by Fickett and Tung [6]. All the sequences are taken from GenBank in May 1992. Fickett and Tung have provided the 21 different coding measures that they surveyed and compared. The benchmark human data includes three different data sets. For the first data set, nonoverlapping human DNA sequences of length 54 have been extracted from all human sequences, with shorter pieces at the ends discarded. Every sequence is labeled according to whether it is entirely coding, entirely noncoding, or mixed, and the mixed sequences (that is, overlapping the exon–intron boundaries) are discarded. The data set also includes the reverse complement of every sequence. This means that one-half of the data is guaranteed to be from the nonsense strand of the DNA, which makes the problem of identifying coding regions somewhat harder. For the current study, the same division into training and test data have been used as in the benchmark study [6]. The training set is used exclusively to construct an MLP-based tree-structured pattern classifier (NNTree), and the tree is then used to classify the test set. In addition to the 54-base data set, the data sets containing 108 and 162 bases have been used. The sizes of these data sets are shown in Table 2.5, which gives the number of nonoverlapping windows in each set. No information about reading frames is used in this study. Every window is either all-coding or all-noncoding, but the reading frame of each window is unknown. This choice of window length and experimental method follows that used by Fickett and Tung [6] and the problem here is what they defined as a protein coding region.

2.4.2 Feature Set

All of the features that have been used are derived from the 21 protein coding measures, which are proposed by Fickett and Tung [6]. A single coding measure is not necessarily the same thing as a single measure. Typically, a coding measure is a vector of measurements on a DNA subsequence.

2.4.2.1 Dicodon Measure

A dicodon is a subsequence of six consecutive nucleotides such as **TAGGAC**. The dicodon measure is the list of the 4,096 frequencies of every possible dicodon (six consecutive bases from the 4 letter alphabet). The dicodon frequency feature is a vector of the 4,096 dicodon frequencies across the input sequence, where the dicodon counts are accumulated only at locations whose starting point is a multiple of 3 (that is, starting at the 0th, 3rd, 6th, . . . nucleotides in the sequence). To convert the dicodon measure to a single number, the 4,096 dicodon frequencies are computed on each window and plugged into the hyperplane equation. This gives a single number that becomes the dicodon discriminant.

2.4.2.2 Hexamer-1 and Hexamer-2 Measures

The hexamer-1 and hexamer-2 measures are identical to dicodons, except that 1 and 2 offsets them. The hexamer-1 frequency feature is likewise a vector of 4,096 dicodon frequencies, except that the counts are accumulated at positions $1, 4, 7, \ldots$.; the hexamer-2 frequency feature is defined analogously.

2.4.2.3 Open Reading Frame Measure

The open reading measure is simply the longest sequence of codons in the window that does not contain a stop codon. That is, the open reading frame feature is the length, in codons, of the longest sequence of codons (aligned with locations 0, 3, . . .) in the data string which does not contain a stop codon.

2.4.2.4 Run Measure

The run measure is a vector of length 14; for each nontrivial subset $S \subset \{\mathbf{A}, \mathbf{C}, \mathbf{T}, \mathbf{G}\}$, the run feature contains an entry that gives the length of the longest contiguous

subsequence having all entries from S. For example, if the entry of the run feature, which corresponds to {**C**, **G**}, is 4, then it means that the longest consecutive substring containing only **C** and **G** is of length 4. The run measure counts the number of repeats or runs of a single base or any set of bases from the set (**A**, **C**, **T**, **G**). Thus, it includes 14 nontrivial subsets of the four bases, and for each subset runs are counted separately.

2.4.2.5 Position Asymmetry Measure

The asymmetry feature is a vector of length four which measures, for each nucleotide **A**, **C**, **G**, and **T**, the extent to which the nucleotide is asymmetrically distributed over the three codon positions. The position asymmetry measure counts for each of the four bases, the frequency of the base in each of the three codon positions. Thus, $f(b, i)$ is the frequency of base b in position i and

$$\mu(b) = \sum_i \frac{f(b, i)}{3}. \tag{2.5}$$

Asymmetry is then defined as

$$\text{asymm}(b) = \sum_i (f(b, i) - \mu(b))^2. \tag{2.6}$$

2.4.2.6 Codon Usage Measure

The codon usage feature is a vector of the 64 codon frequencies. The codon usage measure is simply the frequencies of the 64 possible codons in the test window. The counts are accumulated only at locations 0, 3, ...

2.4.2.7 Diamino Acid Usage Measure

The diamino acid frequency is a vector of the 441 amino acid frequencies which are obtained by translating from the nucleotide sequence to an amino acid string (stop codons are treated as a 21st amino acid); like the dicodon frequency feature, counts are accumulated only at locations 0, 3, ...

2.4.2.8 Fourier Measure

Let $E(x, y)$ be the equality predicate that has value 1 if $x = y$ and 0 otherwise. The nth Fourier coefficient for a window W of length $2M$ is then defined as:

Table 2.6 Human DNA 54bp at $\eta = 0.50$ and $\alpha = 0.70$

Depth of tree	$H_n = 10$			$H_n = 15$		
	Training	Testing	Breadth	Training	Testing	Breadth
1	57.2	54.7	1	57.5	54.9	1
2	84.6	82.5	2	84.5	81.8	2
3	85.2	82.6	4	85.4	82.0	4
4	86.0	82.7	8	86.0	82.1	8
5	86.5	82.9	16	86.4	82.2	16
6	86.5	82.9	16	86.7	82.4	32

$$F(n) = \sum_{p} \sum_{m} E(W_m, W_{m-p}) \exp\left(\frac{\Pi inp}{M}\right) \tag{2.7}$$

where W_m represents the mth base in the window. The Fourier measure is then just $F(2M/2)$, $F(2M/3)$, ..., $F(2M/9)$, which corresponds to the Fourier coefficients for periods 2 through 9.

After generating all protein coding measures, all the attributes in a data set are normalized to facilitate the NNTree learning. Suppose, the possible value range of an attribute $\mathscr{A}_i$ is $(\mathscr{A}_{i,\min}, \mathscr{A}_{i,\max})$, and the real value that class element j takes at $\mathscr{A}_i$ is $\mathscr{A}_{ij}$, then the normalized value of $\mathscr{A}_{ij}$ is given as follows:

$$\overline{\mathscr{A}}_{ij} = \frac{\mathscr{A}_{ij} - \mathscr{A}_{i,\min}}{\mathscr{A}_{i,\max} - \mathscr{A}_{i,\min}}. \tag{2.8}$$

Next subsection presents extensive experimental analysis regarding the classification accuracy of the NNTree, an MLP-based tree-structured classifier.

2.4.3 Experimental Results

In this subsection, the results of the NNTree for three Fickett and Tung's data sets are presented. Values are given for the percentage accuracy on both training and test set. Results of the NNTree on each of the data set are given in Tables 2.6, 2.7. 2.8, 2.9, 2.10, and 2.11. The mean accuracy of training and testing confirm that the evolved NNTree can generalize the data sets presented in Table 2.5 irrespective of the number of attributes, tuples, α, η, and H_n.

In case of Fickett and Tung database, for $L \leq 6$, the values of β_i for all possible nodes or locations of the NNTree are less than ε. So, all the nodes are intermediate or nonterminal nodes for $L \leq 6$. Hence, the NNTree has been grown by splitting all these nonterminal nodes. At $L = 7$, though the value of $\beta_i < \varepsilon$ for each nonterminal node, the training samples of two classes in each nonterminal node are highly correlated. So, at $L = 7$, an MLP cannot be found, corresponding to an intermediate node,

Table 2.7 Human DNA 54bp at $\eta = 0.70$ and $\alpha = 0.70$

Depth of tree	$H_n = 10$			$H_n = 15$		
	Training	Testing	Breadth	Training	Testing	Breadth
1	55.6	53.2	1	53.6	50.2	1
2	77.3	75.5	2	76.1	71.3	2
3	80.0	78.0	4	83.3	78.7	4
4	82.2	79.9	8	85.2	82.4	8
5	83.5	80.9	16	85.3	82.4	16
6	84.4	81.3	32	85.7	82.6	32

Table 2.8 Human DNA 108bp at $\eta = 0.50$ and $\alpha = 0.70$

Depth of tree	$H_n = 10$			$H_n = 15$		
	Training	Testing	Breadth	Training	Testing	Breadth
1	58.4	55.7	1	59.0	56.1	1
2	86.8	82.5	2	87.1	81.0	2
3	88.3	82.6	4	87.8	81.8	4
4	89.6	82.7	8	89.3	82.9	8
5	90.2	82.8	16	90.1	83.5	16
6	90.7	83.1	32	92.2	83.5	32

Table 2.9 Human DNA 108bp at $\eta = 0.70$ and $\alpha = 0.70$

Depth of tree	$H_n = 10$			$H_n = 15$		
	Training	Testing	Breadth	Training	Testing	Breadth
1	55.3	52.5	1	57.4	55.0	1
2	78.9	76.3	2	77.6	74.4	2
3	82.5	79.7	4	85.2	79.3	4
4	84.9	81.9	8	90.8	82.7	8
5	86.5	82.6	16	93.5	82.9	16
6	87.7	83.4	32	93.7	83.5	32

Table 2.10 Human DNA 162bp at $\eta = 0.50$ and $\alpha = 0.70$

Depth of Tree	$H_n = 10$			$H_n = 15$		
	Training	Testing	Breadth	Training	Testing	Breadth
1	59.1	56.9	1	61.0	57.5	1
2	83.5	77.3	2	81.1	71.5	2
3	85.1	78.8	4	84.9	79.6	4
4	88.0	82.9	8	89.9	83.7	8
5	91.4	84.2	16	91.2	84.3	16
6	91.7	84.4	32	91.3	84.3	32

Table 2.11 Human DNA 162bp at $\eta = 0.70$ and $\alpha = 0.70$

Depth of tree	$H_n = 10$			$H_n = 15$		
	Training	Testing	Breadth	Training	Testing	Breadth
1	55.2	53.3	1	58.3	52.8	1
2	82.8	72.5	2	77.8	70.1	2
3	88.1	77.6	4	85.9	76.8	4
4	90.9	83.9	8	89.9	82.3	8
5	93.1	84.2	16	92.7	84.0	16
6	93.1	84.2	32	93.2	84.2	32

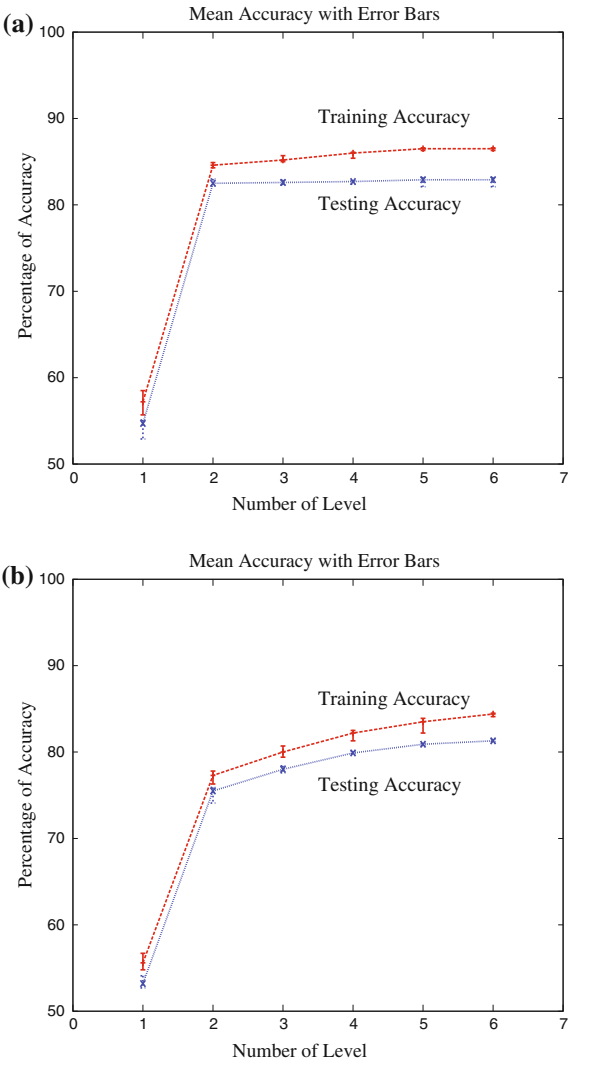

Fig. 2.4 Performance of NNTree on 54bp human DNA sequence for $\alpha = 0.70$ and $H_n = 10$. **a** $\eta = 0.50$; **b** $\eta = 0.70$

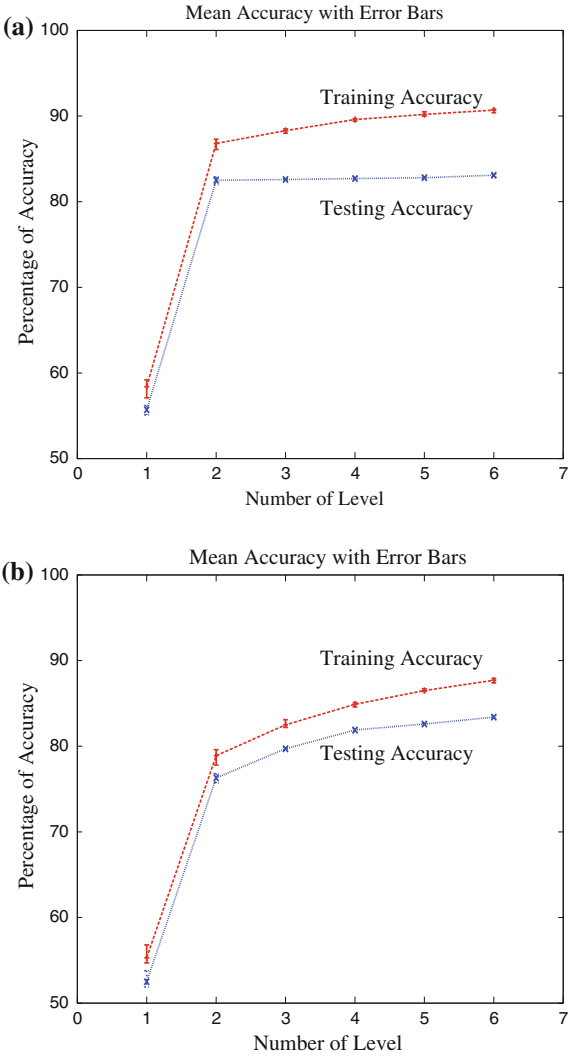

Fig. 2.5 Performance of NNTree on 108bp human DNA sequence for $\alpha = 0.70$ and $H_n = 10$. **a** $\eta = 0.50$; **b** $\eta = 0.70$

which can classify the training samples. So, the learning process terminates at this stage, and the nodes are considered as leaf nodes indicating the class that has the maximum number of training examples in the current location.

Figures 2.4, 2.5, and 2.6 show the classification accuracy with error bar of the NNTree on different DNA sequences. All the results reported in Figs. 2.4, 2.5, and 2.6 establish the fact that the NNTree can generalize a DNA sequence data set irrespective of its sequence length. Also, the standard deviations of training and testing accuracy are very small.

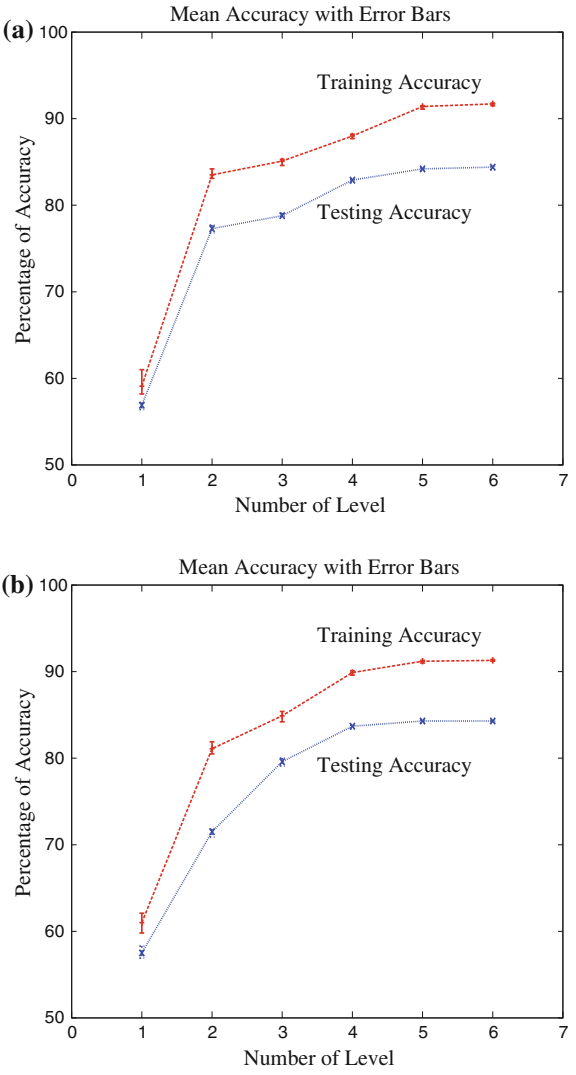

Fig. 2.6 Performance of NNTree on 162bp human DNA sequence for $\eta = 0.50$ and $\alpha = 0.70$. **a** $H_n = 10$; **b** $H_n = 15$

Finally, Table 2.12 compares the classification accuracy of the NNTree with that of OC1 [15, 16], MLP, and other related algorithms. The OC1, proposed by Murty et al. [15, 16], is an oblique decision tree algorithm that combined deterministic hill-climbing with two forms of randomization to find a good oblique split at each intermediate node of a decision tree. All the results reported in Table 2.12 establish the fact that the classification accuracy of the NNTree is higher than that of existing algorithms. Also, the results reported here establish the fact that the NNTree can generalize a DNA data set irrespective of its sequence length.

Table 2.12 Classification accuracy for human DNA 54, 108, and 162bp

Algorithms	54bp	108bp	162bp
NNTree	82.9	83.5	84.4
OC1	73.9	83.7	84.2
MLP	54.9	56.1	57.5
Position asymmetry	70.7	77.6	81.7
Fourier	69.5	77.4	82.0
Hexamer	69.8	71.4	73.8
Dicodon usage	69.8	71.2	73.7

2.5 Conclusion and Discussion

This chapter presents the design of a hybrid learning algorithm, termed as an NNTree. It uses MLP for designing a tree-structured pattern classifier. Instead of using the information gain ratio as a splitting criterion, a new criterion is presented in this chapter for the NNTree design. This criterion captures well the intuitive goal of reducing the rate of misclassification.

The performance of the NNTree is evaluated through its applications in splice-junction and protein coding region identification. Experimental comparisons with other related algorithms provide better or comparable classification accuracy with significantly smaller trees and fast classification times. Extensive experimental results reported in this chapter confirm that the NNTree is crucial over conventional techniques for classification. Also, the sizes of the trees produced by both C4.5 and NNTree have been compared in terms of total number of nodes and height of the trees. A smaller tree is desirable since it provides more compact class descriptions, unless the smaller tree size leads to a loss in accuracy. The results show that the NNTree achieves trees that are significantly smaller than the trees generated by the C4.5.

However, both DNA and protein sequences are nonnumeric variables as they are strings of nucleotides and amino acids, respectively. Hence, for most pattern recognition algorithms, they cannot be used as direct inputs. They, therefore, have to be encoded prior to input. To convert a DNA sequence into numeric values, two methods are reported in this chapter: one is distributed encoding method [17] and the other one is the feature extraction method proposed by Fickett and Tung [6]. In the next chapter, a new encoding method is reported to encode the DNA or protein sequences into numeric values for directly applying different pattern recognition algorithms on them.

References

1. Blaisdell BE (1983) A prevalent persistent global nonrandomness that distinguishes coding and non-coding eucaryotic nuclear dna sequence. J Mol Evol 19(2):122–133
2. Breathnach RJ, Mandel JL, Chambon P (1977) Ovalbumin gene is split in chicken DNA. Nature 270:314–319

3. Cheeseman P, Stutz J (1996) Bayesian classification (AutoClass): theory and results. In: Fayyad UM, Piatetsky-Shapiro G, Smith P, Uthurusamy R (eds) Advances in knowledge discovery and data mining. AAAI/MIT Press, Cambridge, pp 153–180
4. Farber R, Lapedes A, Sirotkin K (1992) Determination of eucaryotic protein coding regions using neural networks and information theory. J Mol Biol 226(2):471–479
5. Fickett J (1982) Recognition of protein coding regions in DNA sequences. Nucleic Acids Res 10(17):5303–5318
6. Fickett J, Tung CS (1992) Assessment of protein coding measures. Nucleic Acids Res 20(24):6441–6450
7. Guo H, Gelfand SB (1992) Classification trees with neural network feature extraction. IEEE Trans Neural Networks 3(6):923–933
8. Hertz J, Krogh A, Palmer RG (1991) Introduction to the theory of neural computation. Addison Wesley, Santa Fe institute studies in the sciences of complexity
9. Koza JR (1994) Genetic programming-II: automatic discovery of reusable programs. MIT Press, Cambridge, ISBN 0262111896
10. Lippmann R (1987) An introduction to computing with neural nets. IEEE Acoust Speech Signal Process Mag 4(2):4–22
11. Maji P (2008) Efficient design of neural network tree using a new splitting criterion. Neurocomputing 71(4–6):787–800
12. Maji P, Das C (2008) Pattern classification using NNtree: design and application for biological data set. J Intell Syst 17(1–3):51–71
13. Maji P, Shaw C, Ganguly N, Sikdar BK, Chaudhuri PP (2003) Theory and application of cellular automata for pattern classification. Fundamenta Informaticae 58:321–354
14. Michie D, Spiegelhalter DJ, Taylor CC (1994) Machine learning, neural and statistical classification. Ellis Horwood, Chichester
15. Murty SK, Kasif S, Salzberg S (1994) A system for identification of oblique decision trees. J Artif Intell Res 2(1):1–32
16. Murty SK, Kasif S, Salzberg S, Beigel R (1993) OC1: randomized induction of oblique decision trees. In: Proceedings of the 11th national conference on artificial intelligence, AAAI/MIT Press, pp 322–327
17. Qian N, Sejnowski TJ (1988) Predicting the secondary structure of globular proteins using neural network models. J Mol Biol 202(4):865–884
18. Quinlan JR (1993) C4.5: programs for machine learning. Morgan Kaufmann, San Fransisco
19. Schmitz GP, Aldrich C, Gouws FS (1999) ANN-DT: an algorithm for extraction of decision trees from artificial neural networks. IEEE Trans Neural Networks 10(6):1392–1401
20. Sethi IK (1990) Entropy nets: from decision trees to neural networks. Proc IEEE 78(10):1605–1613
21. Sethi IK, Yoo JH (1997) Structure-driven induction of decision tree classifiers through neural learning. Pattern Recogn 30(11):1893–1904
22. Song HH, Lee SW (1998) A self-organizing neural network tree for large-set pattern classification. IEEE Trans Neural Networks 9(6):369–380
23. Tay ALP, Zurada JM, Wong LP, Xu J (2007) The hierarchical fast learning artificial neural network (hieflann): an autonomous platform for hierarchical neural network construction. IEEE Trans Neural Networks 18(6):1645–1657
24. Tsukimoto H (2000) Extracting rules from trained neural networks. IEEE Trans Neural Networks 11(2):377–389
25. Uberbacher E, Mural R (1991) Locating protein-coding regions in human dna sequences by a multiple sensor-neural network approach. Proc Nat Acad Sci USA 88(24):11,261–11,265
26. Wilamowski BM, Yu H (2010) Neural network learning without backpropagation. IEEE Trans Neural Networks 21(11):1793–1803
27. Zhao QF (2000) Neural network tree: integration of symbolic and nonsymbolic approaches. Technical Report of IEICE

28. Zhao QF (2001) Evolutionary design of neural network tree-integration of decision tree, neural network and GA. In: Proceedings of the IEEE congress on evolutionary computation, pp. 240–244
29. Zhao QF (2001) Training and retraining of neural network trees. In: Proceedings of the INNS IEEE international joint conference on neural networks, pp. 726–731
30. Zhou ZH, Chen ZQ (2002) Hybrid decision tree. Knowl-Based Syst 15(8):515–528

Chapter 3
Design of String Kernel to Predict Protein Functional Sites Using Kernel-Based Classifiers

3.1 Introduction

The prediction of functional sites in proteins is another important problem in bioinformatics. It is an important issue in protein function studies and hence, drug design. As a result, most researchers use protein sequences for the analysis or the prediction of protein functions in various ways [6, 37]. Thus, one of the major tasks in bioinformatics is the classification and prediction of protein sequences. There are two types of analysis of protein sequences. The first is to analyze whole sequences aiming to annotate novel proteins or classify proteins. In this method, the protein function is annotated through aligning a novel protein sequence with a known protein sequence. If the similarity between a novel sequence and a known sequence is very high, the novel protein is believed to have the same or similar function as the known protein. The second is to recognize functional sites within a sequence. The latter normally deals with subsequences [37].

The problem of functional sites prediction deals with the subsequences; each subsequence is obtained through moving a fixed length sliding window residue by residue. The residues within a scan form a subsequence. If there is a functional site within a subsequence, the subsequence is labeled as functional, otherwise it is labeled as nonfunctional. Therefore, protein subsequence analysis problem is to classify a subsequence whether it is functional or nonfunctional [37]. The major objective in classification analysis is to train a classification model based on the labeled data. The trained model is then used for classifying novel data. Classification analysis requires two descriptions of an object: one is the set of features that are used as inputs to train the model and the other is referred to as the class label. Classification analysis aims to find a mapping function from the features to the class label.

To analyze protein sequences, BLAST [3], suffix-tree-based algorithms [1], regular expression matching representations [34], and finite state machines [30, 31] are a few of the many pattern recognition algorithms that use characters or strings as their primitive type. However, some other pattern recognition algorithms, such as artificial neural networks trained with back-propagation [9, 26, 28], Kohonen's

P. Maji and S. Paul, *Scalable Pattern Recognition Algorithms*,
DOI: 10.1007/978-3-319-05630-2_3,

self-organizing map [4], feed-forward and recurrent neural networks [6, 7], bio-basis function (BBF) neural networks [8, 35, 39–42], and support vector machines [10, 25, 37], work with numerical inputs to predict different functional sites in proteins such as protease cleavage sites of human immunodeficiency virus (HIV) and Hepatitis C Virus, linkage sites of glycoprotein, enzyme active sites, posttranslational phosphorylation sites, immunological domains, Trypsin cleavage sites, protein-protein interaction sites, and so forth. Hence, in order to apply the powerful kernel-based pattern recognition algorithms such as support vector machines to predict functional sites in proteins, the biological data, therefore, have to be encoded prior to input. The objective of coding biological information in sequences is to provide a method for converting nonnumerical attributes in sequences to numerical attributes.

There are two main methods for coding a subsequence, distributed encoding technique [28] and BBF method [8, 35, 41]. The most commonly used method in subsequence analysis is distributed encoding, which encodes each of the 20 amino acids using a 20-bit binary vector [28]. In this method, the input space for modeling is expanded unnecessarily [26]. Also, the ratio of the number of parameters in a model is significantly decreased, which degenerates the statistical significance of the available data for modeling. Moreover, the use of the Euclidean distance may not be able to encode biological content in sequences efficiently [26].

In this background, the concept of BBF has been proposed in [8, 35, 41] for analyzing biological subsequences. The BBF is a string kernel function that transforms nonnumerical biological subsequences to the numerical feature vectors. Transformation of an input biological subsequence to a numerical feature vector is performed based on the similarity of the input subsequence and a set of reference strings. These reference strings are termed as the bio-basis strings and the similarity is calculated using an amino acid mutation matrix. The bio-basis strings are the sections of biological sequences that are used for the transformation of biological data into a numerical feature space with dimension equal to the number of bio-basis strings. The BBF has been successfully applied to predict different functional sites in proteins [8, 35, 39–42].

The most important issue for BBF is how to select a reduced set of most relevant and nonredundant bio-basis strings. Berry et al. [8] used the Fisher ratio for selection of bio-basis strings. Yang and Thomson [41] proposed a method to select bio-basis strings using mutual information. In principle, the bio-basis strings in nonnumerical sequence space should be such that the degree of resemblance between pairs of bio-basis strings would be as minimum as possible. Each of them would then represent a unique feature in numerical feature space. However, the methods proposed in [8, 41] have not adequately addressed this problem. Also, it has not been paid much attention earlier. Moreover, the BBF proposed in [8, 35, 41] does not take into account the impact or influence of each bio-basis string in nonnumerical sequence space. Recently, Maji and Pal [20] proposed a relational clustering algorithm, termed as rough-fuzzy *c*-medoids, to select a set of bio-basis strings for amino acid sequence analysis. The comparative performance analysis of different relational clustering algorithms on bio-basis string selection problem is also reported in [19, 23].

In this chapter, a new string kernel function, termed as novel BBF [21, 22], is reported. It modifies existing BBF and is developed based on the principle of asymmetricity of biological dissimilarity. The dissimilarity is measured using an amino acid mutation matrix. The concept of zone of influence of the bio-basis string is introduced in the novel kernel function to normalize the asymmetric dissimilarity. It takes into account the influence or impact of each bio-basis string in nonnumerical sequence space. An efficient method, introduced in [21] is presented, which integrates the Fisher ratio and the concept of degree of resemblance to select most relevant and distinct bio-basis strings for the novel string kernel function. Instead of using symmetric similarity measure, the asymmetric biological dissimilarity is used to calculate the Fisher ratio, which is more effective for selection of most relevant bio-basis strings. The degree of resemblance enables efficient selection of a set of distinct bio-basis strings. In effect, it reduces the redundant features in numerical feature space. Some quantitative measures are presented to evaluate the quality of selected bio-basis strings. The effectiveness of the novel string kernel function and the new bio-basis string selection method, along with a comparison with existing BBF and related bio-basis string selection methods, is demonstrated on different protein data sets.

The structure of the rest of this chapter is as follows: Sect. 3.2 briefly introduces necessary notions of BBF, and the bio-basis string selection methods proposed by Berry et al. [8] and Yang and Thomson [41]. In Sect. 3.3, a novel string kernel function is presented, integrating the concepts of asymmetricity of biological dissimilarity and the zone of influence of bio-basis string. In Sect. 3.4, an efficient bio-basis string selection method is reported based on the Fisher ratio and the degree of resemblance. Some quantitative performance measures are presented in Sect. 3.5 to evaluate the quality of selected bio-basis strings. A few case studies and a comparison with existing string kernel function and related methods are presented in Sect. 3.6. Concluding remarks are given in Sect. 3.7.

3.2 String Kernel for Protein Functional Site Identification

In this section, the basic notion in the theory of bio-basis function is reported, along with the bio-basis string selection methods proposed by Yang and Thomson [41] and Berry et al. [8].

3.2.1 Bio-Basis Function

A widely used method in sequence analysis is the sequence alignment [2, 3]. In this method, the function of a sequence is annotated through aligning a novel sequence with known sequences. If the alignment between a novel sequence and a known sequence gives a very high similarity (homology) score, the novel sequence is

Table 3.1 Dayhoff mutation matrix: 1 point mutation is accepted per 100 residues (PAM1)

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	40	24	32	32	16	36	28	28	28	24	28	32	36	32	24	36	36	32	8	20
C	24	80	12	12	16	20	20	24	12	8	12	16	20	12	16	32	24	24	0	32
D	32	12	48	44	8	36	36	24	32	16	20	40	28	40	28	32	32	24	4	16
E	32	12	44	48	12	32	36	24	32	20	24	36	28	40	28	32	32	24	4	16
F	16	16	8	12	68	12	24	36	12	40	32	16	12	12	16	20	20	28	32	60
G	36	20	36	32	12	52	24	20	24	16	20	32	28	28	20	36	32	28	4	12
H	28	20	36	36	24	24	56	24	32	24	24	40	32	44	40	28	28	24	20	32
I	28	24	24	24	36	20	24	52	24	40	40	24	24	24	24	28	32	48	12	28
K	28	12	32	32	12	24	32	24	52	20	32	36	28	36	44	32	32	24	20	16
L	24	8	16	20	40	16	24	40	20	56	48	20	20	24	20	20	24	40	24	28
M	28	12	20	24	32	20	24	40	32	48	56	24	24	28	32	24	28	40	16	24
N	32	16	40	36	16	32	40	24	36	20	24	40	28	36	32	36	32	24	16	24
P	36	20	28	28	12	28	32	24	28	20	24	28	56	32	32	36	32	28	8	12
Q	32	12	40	40	12	28	44	24	36	24	28	36	32	48	36	28	28	24	12	16
R	24	16	28	28	16	20	40	24	44	20	32	32	32	36	56	32	28	24	40	16
S	36	32	32	32	20	36	28	28	32	20	24	36	36	28	32	40	36	28	24	20
T	36	24	32	32	20	32	28	32	32	24	28	32	32	28	28	36	44	32	12	20
V	32	24	24	24	28	28	24	48	24	40	40	24	28	24	24	28	32	48	8	24
W	8	0	4	4	32	4	20	12	20	24	16	16	8	12	40	24	12	8	100	32
Y	20	32	16	16	60	12	32	28	16	28	24	24	12	16	16	20	20	24	32	72

believed to have the same or similar function as the known sequence. In this method, an amino acid mutation matrix is commonly used. Each mutation matrix has 20 columns and 20 rows. A value at the nth row and mth column is a probability or a likelihood value that the nth amino acid mutates to the mth amino acid after a particular evolutionary time [15, 17]. The mutation probabilities as similarities among amino acids are, therefore, metrics. The Dayhoff matrix (Table 3.1) was the first mutation matrix developed in 1978 [13] and many new mutation matrices were developed later on, for instance, the Blosum62 matrix [15]. However, the above method may not be useful directly for subsequence analysis. Because, a subsequence may not contain enough information for conventional alignment.

To alleviate this problem, the concept of BBF is introduced in [8, 35, 41] for subsequence analysis, which is based on the principle of conventional alignment technique. Using a table look-up technique, a homology score as a similarity value can be obtained for a pair of subsequences. The nongapped pairwise alignment technique is used to calculate this similarity value, where no deletion or insertion is used to align two subsequences [8, 35, 41]. For ease of discussions, in rest of the chapter, the following terminology is used.

- $\mathbb{A} = \{\text{A}, \text{C}, \ldots, \text{W}, \text{Y}\}$ be the set of 20 amino acids.
- n represents the total number of subsequences.
- $X = \{x_1, \ldots, x_j, \ldots, x_n\}$ be the set of n subsequences with m residues, $\forall x_j \in \mathbb{A}^m$.
- c represents the total number of bio-basis strings.

- $V = \{v_1, \ldots, v_i, \ldots, v_c\}$ be the set of c bio-basis strings and $\forall v_i \in X$.
- $x_j[k] \in \mathbb{A}$, $v_i[k] \in \mathbb{A}$, $\forall_{k=1}^{m}$.

The definition of BBF is as follows [35, 41]:

$$f(x_j, v_i) = \exp\left\{\gamma_b \frac{h(x_j, v_i) - h(v_i, v_i)}{h(v_i, v_i)}\right\} \tag{3.1}$$

- $h(x_j, v_i)$ is the homology score between a subsequence x_j and a bio-basis string v_i calculated using an amino acid mutation matrix [8, 35, 41];
- $h(v_i, v_i)$ denotes the maximum homology score of the ith bio-basis string v_i; and
- γ_b is a constant and typically chosen to be 1 [8, 35].

Suppose both x_j and v_i have m residues, the homology score between x_j and v_i is then defined as

$$h(x_j, v_i) = \sum_{k=1}^{m} \mathbb{M}(x_j[k], v_i[k]) \tag{3.2}$$

where $\mathbb{M}(x_j[k], v_i[k])$ can be obtained from an amino acid mutation matrix through a table look-up method. Note that $x_j[k], v_i[k] \in \mathbb{A}$ and $\mathbb{A}$ is a set of 20 amino acids (Table 3.1).

Consider two bio-basis strings $v_1 =$ KPRT and $v_2 =$ YKAE, and a subsequence $x_1 =$ IPRS having $m = 4$ residues. The nongapped pairwise homology score is calculated between the subsequence x_1 and each bio-basis string considering the mutation probabilities as in Table 3.1. For the first bio-basis string v_1, four mutation probabilities are

$$\mathbb{M}(x_1[1], v_1[1]) = \mathbb{M}(\text{I, K}) = 24; \quad \mathbb{M}(x_1[2], v_1[2]) = \mathbb{M}(\text{P, P}) = 56;$$
$$\mathbb{M}(x_1[3], v_1[3]) = \mathbb{M}(\text{R, R}) = 56; \quad \mathbb{M}(x_1[4], v_1[4]) = \mathbb{M}(\text{S, T}) = 36.$$

Hence, the homology score between the subsequence x_1 and the bio-basis string v_1 is given by

$$h(x_1, v_1) = \sum_{k=1}^{4} \mathbb{M}(x_1[k], v_1[k]) = 172.$$

Similarly, for the second bio-basis string v_2, four mutation probabilities are 28, 28, 24, and 32. Thus, the value of $h(x_1, v_2)$ between the subsequence x_1 and the bio-basis string v_2 is as follows:

$$h(x_1, v_2) = \sum_{k=1}^{4} \mathbb{M}(x_1[k], v_2[k]) = 112.$$

The maximum homology scores of two bio-basis strings v_1 and v_2 are given by

$$h(v_1, v_1) = 208 \quad \text{and} \quad h(v_2, v_2) = 212.$$

Considering the value of $\gamma_b = 1$

$$f(x_1, v_1) = \exp\left\{\gamma_b \frac{h(x_1, v_1) - h(v_1, v_1)}{h(v_1, v_1)}\right\} = 0.633334;$$
$$f(v_1, v_1) = \exp\left\{\gamma_b \frac{h(v_1, v_1) - h(v_1, v_1)}{h(v_1, v_1)}\right\} = 1.000000;$$
$$f(x_1, v_2) = \exp\left\{\gamma_b \frac{h(x_1, v_2) - h(v_2, v_2)}{h(v_2, v_2)}\right\} = 0.287988;$$
$$f(v_2, v_2) = \exp\left\{\gamma_b \frac{h(v_2, v_2) - h(v_2, v_2)}{h(v_2, v_2)}\right\} = 1.000000.$$

Hence, the value of BBF $f(x_i, x_j)$ is high if two subsequences x_i and x_j are similar or close to each other. The function value is one if two subsequences are identical, and small if they are distinct. The function needs a subsequence as a support (bio-basis string). Each bio-basis string is a feature dimension in a numerical feature space. If $\mathbb{A}$ is used to denote a collection of 20 amino acids, an input space of all potential subsequences with m residues is $\mathbb{A}^m$. Then, a collection of c bio-basis strings formulates a numerical feature space $\mathbb{R}^c$, to which a nonnumerical sequence space $\mathbb{A}^m$ is mapped for analysis. More importantly, the BBF can transform various homology scores to a real number as a similarity within the interval [0, 1], that is,

$$0 \leq f(x_j, v_i) \leq 1. \tag{3.3}$$

After the mapping using BBF, a nonnumerical subsequence space $\mathbb{A}^m$ will be mapped to a c-dimensional numerical feature space $\mathbb{R}^c$, that is, $\mathbb{A}^m \rightarrow \mathbb{R}^c$.

3.2.2 Selection of Bio-Basis Strings Using Mutual Information

In [41], Yang and Thomson proposed a method for bio-basis string selection using mutual information [32]. The necessity for a bio-basis string to be an independent and informative feature can be determined by the shared information between the bio-basis string and the rest as well as the shared information between the bio-basis string and class label [41].

The mutual information is quantified as the difference between the initial uncertainty and the conditional uncertainty. Let $\Phi = \{v_i\}$ be a set of selected bio-basis strings, $\Theta = \{v_k\}$ a set of candidate bio-basis strings. $\Phi = \phi$ (empty) at the beginning. A prior probability of a bio-basis string v_k is referred as $p(v_k)$. The initial uncertainty of v_k is defined as

$$H(v_k) = -p(v_k) \ln p(v_k). \tag{3.4}$$

Similarly, the joint entropy of two bio-basis strings v_k and v_i is given by

$$H(v_k, v_i) = -p(v_k, v_i) \ln p(v_k, v_i) \tag{3.5}$$

where $v_i \in \Phi$ and $v_k \in \Theta$. The mutual information between v_k and v_i is, therefore, given by

$$\begin{aligned} I(v_k, v_i) =& H(v_k) + H(v_i) - H(v_k, v_i) \\ =& \{-p(v_k) \ln p(v_k) - p(v_i) \ln p(v_i) \\ &+ p(v_k, v_i) \ln p(v_k, v_i)\}. \end{aligned} \tag{3.6}$$

However, the mutual information of v_k with respect to all the bio-basis strings in Φ is

$$I(v_k, \Phi) = \sum_{v_i \in \Phi} I(v_k, v_i). \tag{3.7}$$

Combining (3.6) and (3.7), we get [41]

$$I(v_k, \Phi) = \sum_{v_i \in \Phi} p(v_k, v_i) \ln \left\{ \frac{p(v_k, v_i)}{p(v_k) p(v_i)} \right\}. \tag{3.8}$$

Replacing Φ with the class label $\Omega = \{\Omega_1, \ldots, \Omega_j, \ldots, \Omega_M\}$, the mutual information

$$I(v_k, \Omega) = \sum_{\Omega_j \in \Omega} p(v_k, \Omega_j) \ln \left\{ \frac{p(v_k, \Omega_j)}{p(v_k) p(\Omega_j)} \right\} \tag{3.9}$$

measures the mutual relationship between v_k and Ω. A bio-basis string whose $I(v_k, \Omega)$ value is the largest will be selected as v_k and will make the largest contribution to modeling (discrimination using Ω) among all the remaining bio-basis strings in Θ. Therefore, there are two mutual information measurements for v_k, the mutual information between v_k and Ω ($I(v_k, \Omega)$) and the mutual information between v_k and Φ ($I(v_k, \Phi)$). In this method, the following criterion is used for the selection of bio-basis strings [38, 41]

$$J(v_k) = \alpha_{\mathrm{YT}} I(v_k, \Omega) - (1 - \alpha_{\mathrm{YT}}) I(v_k, \Phi) \tag{3.10}$$

where α_{YT} is a constant. In the current study, the value of α_{YT} is set at 0.7 to give more weightage in discrimination [38, 41]. The major drawback of the method proposed by Yang and Thomson [41] is a huge number of prior and joint probabilities are to be calculated, which makes the method computationally expensive.

3.2.3 Selection of Bio-Basis Strings Using Fisher Ratio

In [8], Berry et al. proposed a method to select a set $V = \{v_1, \ldots, v_i, \ldots, v_c\}$ of c bio-basis strings from the whole set $X = \{x_1, \ldots, x_j, \ldots, x_n\}$ of n subsequences based on their discriminant capability. The discriminant capability of each subsequence is calculated using the Fisher ratio [14]. The Fisher ratio is used to maximize the discriminant capability of a subsequence in terms of interclass separation (as large as possible) and intraclass spread between subsequences (as small as possible). The larger the Fisher ratio value, the larger the discriminant capability of the subsequence. Based on the values of the Fisher ratio, n subsequences of X can be ranked from the strongest discriminant capability to the weakest one. The method yields a set V of c subsequences from X as the bio-basis strings which possess good discriminant capability between two classes, having evolved from original data set.

However, the n subsequences of X would have different compositions of amino acids. Hence, they should have different pairwise alignment scores with the other subsequences of X. As the class properties of these training subsequences are known, these similarity values can be partitioned into two groups or classes (functional and nonfunctional), which are denoted as $X_A \subset X$ and $X_B \subset X$, respectively. Denoting the similarity between two subsequences x_i and x_j as $h(x_j, x_i)$, the mean and standard deviation values for these two groups with respect to the subsequence x_i are as follows:

$$U_{A_i} = E_A[h(x_j, x_i)] = \frac{1}{n_A} \sum h(x_j, x_i); \qquad \forall x_j \in X_A \tag{3.11}$$

$$U_{B_i} = E_B[h(x_k, x_i)] = \frac{1}{n_B} \sum h(x_k, x_i); \qquad \forall x_k \in X_B \tag{3.12}$$

$$\begin{aligned} \sigma^2_{A_i} &= E_A[h^2(x_j, x_i)] - [E_A[h(x_j, x_i)]]^2 \\ &= \frac{1}{n_A} \sum \{h(x_j, x_i) - U_{A_i}\}^2; \quad \forall x_j \in X_A \end{aligned} \tag{3.13}$$

$$\begin{aligned} \sigma^2_{B_i} &= E_B[h^2(x_k, x_i)] - [E_B[h(x_k, x_i)]]^2 \\ &= \frac{1}{n_B} \sum \{h(x_k, x_i) - U_{B_i}\}^2; \quad \forall x_k \in X_B \end{aligned} \tag{3.14}$$

where n_A and n_B are the number of similarity values in X_A and X_B, respectively. $E[h(x_j, x_i)]$ and $E[h^2(x_k, x_i)]$ represent the zero-mean, first and second order moment of similarity, that is, expectation of $h(x_j, x_i)$ and $h^2(x_j, x_i)$, respectively. Based on these four quantities, the discriminant capability of each subsequence can be measured using the Fisher ratio

$$F(x_i) = \frac{|U_{A_i} - U_{B_i}|}{\sqrt{\sigma_{A_i}^2 + \sigma_{B_i}^2}} \tag{3.15}$$

where

$$|U_{A_i} - U_{B_i}| = |E_A[h(x_j, x_i)] - E_B[h(x_k, x_i)]|; \tag{3.16}$$

and

$$\sigma_{A_i}^2 + \sigma_{B_i}^2 = \{E_A[h^2(x_j, x_i)] + E_B[h^2(x_k, x_i)]\} - \{[E_A[h(x_j, x_i)]]^2 + [E_B[h(x_k, x_i)]]^2\}. \tag{3.17}$$

The basic steps of this method follows next:

1. Calculate the discriminant capabilities of all subsequences of X using the Fisher ratio as in (3.15).
2. Rank all subsequences of X based on the values of Fisher ratio in descending order.
3. Select first c subsequences from X as the set V of bio-basis strings.

However, the bio-basis strings in nonnumerical sequence space should be such that the similarity between pairs of bio-basis strings would be as minimum as possible. Each of them would then represent a unique feature in numerical feature space. The methods proposed in [8, 41] have not adequately addressed this problem. Also, not much attention has been paid to it earlier.

3.3 Novel String Kernel Function

In this section, a novel string kernel function is presented [21] based on the concepts of biological dissimilarity and zone of influence of bio-basis string. Next, an efficient method is reported for selection of bio-basis strings integrating the Fisher ratio and the principle of degree of resemblance.

3.3.1 Asymmetricity of Biological Dissimilarity

Here, we define two asymmetric dissimilarities between two subsequences x_i and x_j as follows [21]:

$$\begin{aligned} d_{x_i \to x_j} &= d(x_j, x_i) = \{h(x_i, x_i) - h(x_j, x_i)\} \\ d_{x_j \to x_i} &= d(x_i, x_j) = \{h(x_j, x_j) - h(x_i, x_j)\} \end{aligned} \tag{3.18}$$

where $d_{x_i \to x_j}$ denotes the dissimilarity of subsequence x_j from the subsequence x_i and $h(x_i, x_j) = h(x_j, x_i)$ is the nongapped homology score between two subsequences x_i and x_j.
Consider two subsequences x_i = KPRT and x_j = YKAE with 4 residues. According to the Dayhoff mutation matrix (Table 3.1), the nongapped pairwise homology score between two subsequences x_i and x_j is, therefore, $h(x_i, x_j) = h(x_j, x_i) = 100$, while the maximum homology scores of two subsequences x_i and x_j are given by $h(x_i, x_i) = 208$ and $h(x_j, x_j) = 212$, respectively. Hence, the dissimilarity of subsequence x_j from subsequence x_i is given by

$$d(x_j, x_i) = \{h(x_i, x_i) - h(x_j, x_i)\} = 208 - 100 = 108,$$

whereas the dissimilarity of x_i from x_j is as follows:

$$d(x_i, x_j) = \{h(x_j, x_j) - h(x_i, x_j)\} = 212 - 100 = 112.$$

Thus, the dissimilarity is asymmetric in nature, that is,

$$d(x_j, x_i) \neq d(x_i, x_j). \tag{3.19}$$

The asymmetricity reflects domain organizations of two subsequences x_i and x_j. When two subsequences x_i and x_j consist of the same single domain, $d(x_j, x_i)$ and $d(x_i, x_j)$ will be similar small values. However, suppose that x_i has one extra domain, then $d(x_j, x_i)$ becomes large even if $d(x_i, x_j)$ is small. These dissimilarities may be used for clustering of protein sequences or subsequences so that domain organizations are well reflected. The asymmetric property of the biological dissimilarity was also observed by Stojmirovic [33] and Itoh et al. [16]. The asymmetric dissimilarity might be a powerful tool to cluster sequences or subsequences and to explore gene/protein universe.

3.3.2 Novel Bio-Basis Function

The design of novel string kernel function is based on the principle of asymmetric biological dissimilarity [21]. Using a table look-up technique, a biological dissimilarity is calculated for a pair of subsequences based on an amino acid mutation matrix. The nongapped pairwise alignment method is used to calculate this dissimilarity, where no deletion or insertion is used to align two subsequences. The definition of the novel bio-basis function (nBBF) is as follows [21]:

$$f_{\text{novel}}(x_j, v_i) = \exp\left\{\frac{\{h(x_j, v_i) - h(v_i, v_i)\}}{\eta_i}\right\}$$

that is,

$$f_{\text{novel}}(x_j, v_i) = \exp\left\{\frac{-d(x_j, v_i)}{\eta_i}\right\} \tag{3.20}$$

The parameter η_i in (3.20) represents the zone of influence of the ith bio-basis string v_i. It represents the variance of the bio-basis string v_i with respect to the subsequences nearest to it. In other words, if each bio-basis string is considered as a cluster prototype, then the zone of influence of it represents the radius of that cluster. The value of η_i could be the same for all bio-basis strings if all of them are expected to form similar clusters in nonnumerical sequence space. In general, it is desirable that η_i should relate to the overall size and shape of the cluster associated with the bio-basis string v_i. In the present research work, the following definition is used:

$$\eta_i = \frac{1}{n_i}\sum_{x_j} d(x_j, v_i) = \frac{1}{n_i}\sum_{x_j}\{h(v_i, v_i) - h(x_j, v_i)\} \tag{3.21}$$

where n_i is the total number of subsequences having minimum dissimilarity from the ith bio-basis string v_i among all the bio-basis strings and $\{h(v_i, v_i) - h(x_j, v_i)\}$ is the dissimilarity of the subsequence x_j from the ith bio-basis string v_i. In other words, the value of η_i represents the average dissimilarity of the input subsequences from their nearest bio-basis string v_i.

Hence, the novel string kernel function normalizes the asymmetric dissimilarity using the zone of influence or variance of the bio-basis string, rather than using maximum homology score of the bio-basis string as in (3.1).

3.4 Biological Dissimilarity Based String Selection Method

One of the main problem in BBF is how to select a reduced set of most relevant bio-basis strings. The bio-basis strings are the sections of biological sequences that are used for the transformation of biological data into a numerical feature space. Hence, the problem of selecting a set $V = \{v_1, \ldots, v_i, \ldots, v_c\}$ of c subsequences as the bio-basis strings from the whole set $X = \{x_1, \ldots, x_j, \ldots, x_n\}$ of n subsequences, where $V \subset X$, is a feature selection problem.

In real biological data analysis, the data set may contain a number of similar or redundant subsequences with low discriminant capability or relevance to the classes. The selection of such similar and nonrelevant subsequences as the bio-basis strings may lead to a reduction in the useful information in numerical feature space. Ideally, the selected bio-basis strings should have high discriminant capability with the classes while the similarity among them would be as low as possible. The subsequences with high discriminant capability are expected to be able to predict the classes of the subsequences. However, the prediction capability may be reduced if many similar

subsequences are selected as the bio-basis strings. In contrast, a data set that contains subsequences not only with high relevance with respect to the classes but with low mutual redundancy is more effective in its prediction capability. Hence, to assess the effectiveness of the subsequences as the bio-basis strings, both the relevance and the redundancy or similarity need to be measured quantitatively. The bio-basis string selection method, reported in [21], addresses the above issues through following three phases:

1. computation of the discriminant capability or relevance of each subsequence;
2. determination of the nonrelevant subsequences; and
3. computation of the similarity or redundancy among subsequences.

An asymmetric biological dissimilarity based Fisher ratio is chosen here to compute the discriminant capability or relevance of each subsequence to the classes, while a novel concept of the degree of resemblance is used to calculate the mutual redundancy or similarity among subsequences. The nonrelevant subsequences are discarded using a nearest mean classifier [14]. Next the calculation of the Fisher ratio using asymmetric biological dissimilarity is provided along with the concept of degree of resemblance and the principle of nearest mean classifier.

3.4.1 Fisher Ratio Using Biological Dissimilarity

In the method reported in [21], the Fisher ratio [14] is used to measure the discriminant capability or relevance of each subsequence $x_i \in X$. The Fisher ratio is calculated based on the asymmetric biological dissimilarity. As the class labels of all training subsequences are known, the set X can be partitioned into two groups or classes X_A and X_B, respectively, where

$$X_A \cap X_B = \emptyset; \qquad X_A \cup X_B = X; \tag{3.22}$$

$$|X_A| = n_A; \qquad |X_B| = n_B; \qquad n_A + n_B = n. \tag{3.23}$$

Hence, each subsequence $x_i \in X$ should have n_A and n_B dissimilarity values with the subsequences of X_A and X_B, respectively. Denoting the dissimilarity of the subsequence x_j from the subsequence x_i as $d(x_j, x_i)$, the mean and standard deviation values for the two classes X_A and X_B with respect to the subsequence x_i are as follows:

$$\overline{U}_{A_i} = \frac{1}{n_A} \sum d^2(x_j, x_i); \qquad \forall x_j \in X_A \tag{3.24}$$

$$\overline{U}_{B_i} = \frac{1}{n_B} \sum d^2(x_k, x_i); \qquad \forall x_k \in X_B \tag{3.25}$$

$$\overline{\sigma}^2_{A_i} = \frac{1}{n_A}\sum\{d^2(x_j, x_i) - \overline{U}_{A_i}\}^2; \qquad \forall x_j \in X_A \tag{3.26}$$

$$\overline{\sigma}^2_{B_i} = \frac{1}{n_B}\sum\{d^2(x_k, x_i) - \overline{U}_{B_i}\}^2; \qquad \forall x_k \in X_B \tag{3.27}$$

where $\overline{U}_{A_i}$, $\overline{U}_{B_i}$, $\overline{\sigma}_{A_i}$, and $\overline{\sigma}_{B_i}$ represent the mean and standard deviation values of the subsequence x_i for two groups X_A and X_B, respectively. These four quantities are calculated based on the square of biological dissimilarity with respect to the subsequence x_i. Based on these four quantities, the discriminant capability of each subsequence x_i is computed using the Fisher ratio that is as follows:

$$\overline{\mathbb{F}}(x_i) = \frac{|\overline{U}_{A_i} - \overline{U}_{B_i}|}{\sqrt{\overline{\sigma}^2_{A_i} + \overline{\sigma}^2_{B_i}}}. \tag{3.28}$$

Let $\kappa_i = h(x_i, x_i)$ be the maximum homology score of the subsequence x_i. The above four quantities can then be written using κ_i as

$$\overline{U}_{A_i} = \{\kappa_i^2 + E_A[h^2(x_j, x_i)] - 2\kappa_i E_A[h(x_j, x_i)]\}; \tag{3.29}$$

$$\overline{U}_{B_i} = \{\kappa_i^2 + E_B[h^2(x_k, x_i)] - 2\kappa_i E_B[h(x_k, x_i)]\}; \tag{3.30}$$

$$\begin{aligned}\overline{\sigma}^2_{A_i} = \{&4\kappa_i^2(E_A[h^2(x_j, x_i)] - [E_A[h(x_j, x_i)]]^2) \\ &- 4\kappa_i(E_A[h^3(x_j, x_i)] - E_A[h(x_j, x_i)]E_A[h^2(x_j, x_i)]) \\ &- [E_A[h^2(x_j, x_i)]]^2 + E_A[h^4(x_j, x_i)]\};\end{aligned} \tag{3.31}$$

and

$$\begin{aligned}\overline{\sigma}^2_{B_i} = \{&4\kappa_i^2(E_B[h^2(x_k, x_i)] - [E_B[h(x_k, x_i)]]^2) \\ &- 4\kappa_i(E_B[h^3(x_k, x_i)] - E_B[h(x_k, x_i)]E_B[h^2(x_k, x_i)]) \\ &- [E_B[h^2(x_k, x_i)]]^2 + E_B[h^4(x_k, x_i)]\};\end{aligned} \tag{3.32}$$

where $E[h^r(x_j, x_i)]$ represents the zero-mean, rth order moment of similarity $h(x_j, x_i)$ between two subsequences x_i and x_j. Now, the numerator of (3.28) is given by

$$\begin{aligned}|\overline{U}_{A_i} - \overline{U}_{B_i}| = |\{&E_A[h^2(x_j, x_i)] - E_B[h^2(x_k, x_i)]\} \\ &- 2\kappa_i\{E_A[h(x_j, x_i)] - E_B[h(x_k, x_i)]\}|.\end{aligned} \tag{3.33}$$

Hence, the numerator of (3.28) not only depends on the difference of zero-mean, first-order moment of similarity of two groups as in (3.15), it also takes into account the difference of zero-mean, second-order moment of similarity as well as the maxi-

mum homology score of the subsequence x_i. That is, the numerator of (3.28) depends on following three factors:

- difference of zero-mean, first order moment of similarity of two groups, $\{E_A[h(x_j, x_i)] - E_B[h(x_k, x_i)]\}$;
- difference of zero-mean, second-order moment of similarity of two groups, $\{E_A[h^2(x_j, x_i)] - E_B[h^2(x_k, x_i)]\}$;
- maximum homology score of the subsequence x_i, that is, $\kappa_i = h(x_i, x_i)$.

Similarly, the denominator of (3.28) contains the following terms:

$$\begin{aligned}\overline{\sigma}^2_{A_i} + \overline{\sigma}^2_{B_i} = {} & [4\kappa_i^2\{E_A[h^2(x_j, x_i)] + E_B[h^2(x_k, x_i)]\} \\ & - 4\kappa_i^2\{[E_A[h(x_j, x_i)]]^2 + [E_B[h(x_k, x_i)]]^2\} \\ & - 4\kappa_i\{E_A[h^3(x_j, x_i)] + E_B[h^3(x_k, x_i)]\} \\ & + 4\kappa_i\{E_A[h(x_j, x_i)]E_A[h^2(x_j, x_i)] \\ & + E_B[h(x_k, x_i)]E_B[h^2(x_k, x_i)]\} \\ & - \{[E_A[h^2(x_j, x_i)]]^2 + [E_B[h^2(x_k, x_i)]]^2\} \\ & + \{E_A[h^4(x_j, x_i)] + E_B[h^4(x_k, x_i)]\}]. \end{aligned} \tag{3.34}$$

Hence, the denominator of (3.28) considers the zero-mean, higher order (upto fourth order) moment of similarity of two groups as well as the maximum homology score of the subsequence x_i, while that of (3.15) only takes into account the zero-mean, first- and second-order moment of similarity of two groups and does not consider the maximum homology score of the subsequence x_i. In effect, (3.28) calculates the discriminant capability of each subsequence x_i more accurately.

3.4.2 Nearest Mean Classifier

After computing the discriminant capability or relevance $\overline{\mathbb{F}}(x_i)$ of each subsequence $x_i \in X$ using the Fisher ratio according to (3.28), the nonrelevant subsequences are discarded based on a threshold value δ. The subsequences those have the Fisher ratio values larger than or equal to the threshold value δ are considered as the candidate bio-basis strings. The value of δ is obtained using the concept of nearest mean classifier.

The current technique assumes at least one bio-basis string in the set X. If the Fisher ratio value $\overline{\mathbb{F}}(x_t)$ of the subsequence x_t is the maximum, then x_t is declared to be the first bio-basis string. In order to find other candidate bio-basis strings, the threshold value δ is calculated using the nearest mean classifier. To obtain the reliable arithmetic mean, the subsequence x_t and those have the Fisher ratio values less than $\overline{\mathbb{F}}(x_t)/10$ are removed [18]. The mean $\mathscr{M}$ of the Fisher ratio values of the remaining subsequences is then calculated. Finally, the minimum mean distance is calculated

as follows:

$$\mathscr{D}(x_s) = \min |\overline{\mathbb{F}}(x_i) - \mathscr{M}| : \qquad 1 < i < n \tag{3.35}$$

where the Fisher ratio value $\overline{\mathbb{F}}(x_s)$ of the subsequence x_s has the minimum distance with $\mathscr{M}$. To make the threshold value noise-insensitive, the Fisher ratio value $\overline{\mathbb{F}}(x_s)$ that is closest to the mean $\mathscr{M}$ is set as δ, rather than the mean itself, that is,

$$\delta = \overline{\mathbb{F}}(x_s). \tag{3.36}$$

The basic steps of this approach follows next:

1. Compute the mean $\mathscr{M}$ of the Fisher ratio values of the subsequences without considering the best and below one tenth best Fisher ratio values.
2. Find out the Fisher ratio $\overline{\mathbb{F}}(x_s)$ of the subsequence x_s that has the minimum distance with $\mathscr{M}$ and set the threshold value $\delta = \overline{\mathbb{F}}(x_s)$.
3. Remove those subsequences with Fisher ratio values below the threshold δ.

After eliminating nonrelevant subsequences using the principle of nearest mean classifier, the redundancy among existing subsequences (candidate bio-basis strings) is calculated in terms of nongapped homology score. A quantitative measure is reported next to compute the similarity or redundancy between two subsequences.

3.4.3 Degree of Resemblance

The degree of resemblance of the subsequence x_j with respect to the subsequence x_i is defined as follows [20, 21]:

$$\mathrm{DOR}(x_j, x_i) = \frac{h(x_j, x_i)}{h(x_i, x_i)}. \tag{3.37}$$

It is the ratio between the nongapped pairwise homology score of two input subsequences x_i and x_j to the maximum homology score of the subsequence x_i. It is used to quantify the similarity in terms of homology score between pairs of subsequences. Combining (3.18) and (3.37), the relation between the degree of resemblance and the asymmetric dissimilarity of the subsequence x_j with respect to the subsequence x_i is

$$d(x_j, x_i) = h(x_i, x_i)[1 - \mathrm{DOR}(x_j, x_i)]. \tag{3.38}$$

Let us consider two subsequences x_i = KPRT and x_j = YKAE with 4 residues. The nongapped homology score between x_i and x_j is given by $h(x_i, x_j) = h(x_j, x_i) = 100$, while the maximum homology scores of two subsequences x_i and x_j are given by $h(x_i, x_i) = 208$ and $h(x_j, x_j) = 212$, respectively. Hence, the degree of resemblance of subsequence x_j with respect to the subsequence x_i is given by

$$\mathrm{DOR}(x_j, x_i) = \frac{h(x_j, x_i)}{h(x_i, x_i)} = \frac{100}{208} = 0.480769,$$

while that of x_i with respect to x_j is as follows:

$$\mathrm{DOR}(x_i, x_j) = \frac{h(x_i, x_j)}{h(x_j, x_j)} = \frac{100}{212} = 0.471698.$$

Hence, the degree of resemblance is asymmetric in nature, that is,

$$\mathrm{DOR}(x_i, x_j) \neq \mathrm{DOR}(x_j, x_i). \tag{3.39}$$

This asymmetric property makes a reference subsequence different from the subsequence under study. It helps to find out redundant subsequences with respect to a selected bio-basis string. If two subsequences are different, the degree of resemblance between them is small. A high value of $\mathrm{DOR}(x_i, x_j)$ between two subsequences x_i and x_j asserts that the similarity between them is high. If two subsequences are same, the degree of resemblance between them is maximum, that is, 1. Thus,

$$0 < \mathrm{DOR}(x_i, x_j) \leq 1. \tag{3.40}$$

3.4.4 Details of the Algorithm

While the Fisher ratio is used to calculate the discriminant capability or relevance of each subsequence, the degree of resemblance takes into account the similarity or redundancy between two subsequences. Based on the concept of degree of resemblance as in (3.37) and the Fisher ratio as in (3.28), the method for selecting a reduced set of most relevant bio-basis strings is described next.

- Input: $X = \{x_1, \ldots, x_j, \ldots, x_n\}$ be the set of n subsequences with m residues, where $x_{jk} \in \mathbb{A}$ and $\mathbb{A} = \{\mathrm{A}, \mathrm{C}, \ldots, \mathrm{W}, \mathrm{Y}\}$ be the set of 20 amino acids.
- Output: $V = \{v_1, \ldots, v_i, \ldots, v_c\}$ be the set of c bio-basis strings with m residues, where $v_i \in X$ and $v_{ik} \in \mathbb{A}$.

1. Initialize $\overline{V} \leftarrow X$ and $V \leftarrow \emptyset$.
2. Calculate the discriminant capabilities of all subsequences of $\overline{V}$ using the Fisher ratio as in (3.28).
3. Compute the threshold value δ using (3.36).
4. Remove the subsequences from $\overline{V}$ those have Fisher ratio values below the threshold δ.
5. Repeat steps (a) and (b) for all the remaining subsequences of $\overline{V}$.
   a. Select a subsequence from $\overline{V}$ as the candidate bio-basis string of V that has the highest Fisher ratio value (maximum discriminant capability).

b. Remove the subsequences from $\overline{V}$ those have the DOR values with respect to the selected bio-basis string of step (a) above threshold ξ.

6. End.

Note that the main motive of introducing the concepts of degree of resemblance and nearest mean classifier lies in reducing the number of bio-basis strings. That is, both attempt to eliminate nonrelevant and redundant bio-basis strings from the whole subsequences. The whole approach is, therefore, data-dependent.

3.4.5 Computational Complexity

The time required to compute the Fisher ratio for each subsequence is $\mathcal{O}(nm)$, where n and m are the total number of subsequences and size of each subsequence, respectively. Hence, the complexity to calculate the Fisher ratio of n subsequences is $\mathcal{O}(n^2m)$. Among n subsequences, c subsequences are selected as the c bio-basis strings. The time required to select a bio-basis string from the available n subsequences based on Fisher ratio is $\mathcal{O}(n)$ and to discard subsequences with the DOR values greater than ξ with respect to the currently selected bio-basis string is also $\mathcal{O}(n)$. These two steps are repeated c times to generate c bio-basis strings. So, the time complexity of this phase is $\mathcal{O}(cn)$. Hence, the overall time complexity of the string selection algorithm is $\mathcal{O}(n^2m + cn)$, that is, $\mathcal{O}(n^2)$, as both $c, m \ll n$.

3.5 Quantitative Measure

In this section, some quantitative indices are reported to evaluate the quality of selected bio-basis strings incorporating the concept of biological dissimilarity. Based on this concept, three indices, α, β, and γ [21], are presented next for evaluating quantitatively the quality of selected bio-basis strings. In this regard, it should be noted that some other quantitative indices are reported in [20, 23] to evaluate the quality of selected bio-basis strings incorporating the concept of nongapped pairwise homology alignment score and mutual information.

3.5.1 Compactness: α Index

It is defined as

$$\alpha = \frac{1}{c}\sum_{i=1}^{c}\frac{1}{n_i}\sum_{x_j}\{h(v_i, v_i) - h(x_j, v_i)\} \tag{3.41}$$

where n_i is the total number of subsequences having minimum dissimilarity values from the ith bio-basis string v_i among all the bio-basis strings and $\{h(v_i, v_i) - h(x_j, v_i)\}$ is the dissimilarity of the subsequence x_j from the bio-basis string v_i. The α index represents the average dissimilarity of the input subsequences from their corresponding bio-basis strings. In other words, the α index measures the compactness of input subsequences with respect to their corresponding bio-basis strings. A good bio-basis string selection method should make all input subsequences as close to their bio-basis strings as possible. The value of α index increases with the increase in dissimilarity of all the subsequences from their corresponding bio-basis strings. Therefore, for a given data set and c value, the lower the average dissimilarity, the lower would be the α value. The α value decreases with the increase in compactness of subsequences with respect to their corresponding bio-basis strings.

3.5.2 Cluster Separability: β Index

The β index symmetrizes two asymmetric dissimilarities between two bio-basis strings so that the difference can be evaluated with a single value. It is defined as the minimum of the dissimilarity values between two bio-basis strings and is as follows:

$$\begin{aligned}\beta &= \min_{i,j}\left\{\frac{1}{2}\{d(v_j, v_i) + d(v_i, v_j)\}\right\} : 1 < i, j < c \\ &= \min_{i,j}\left\{\frac{1}{2}\{h(v_i, v_i) + h(v_j, v_j) - 2h(v_j, v_i)\}\right\}. \end{aligned} \tag{3.42}$$

A good bio-basis string selection procedure should make the asymmetric dissimilarity between all bio-basis strings as high as possible. In other words, the β index measures how the bio-basis strings are separated from each other. The β index increases with the increase of dissimilarity between bio-basis strings.

3.5.3 Class Separability: γ Index

It is defined as

$$\gamma = \min_i \left\{\acute{U}_{A_i} - \acute{U}_{B_i}\right\} : \qquad 1 < i < c \tag{3.43}$$

where $\acute{U}_{A_i}$ and $\acute{U}_{B_i}$ are the average dissimilarity values of the ith bio-basis string v_i from all bio-basis strings of two classes Ω_A and Ω_B, respectively and are as follows:

$$\acute{U}_{A_i} = \frac{1}{c_1}\sum_{j=1}^{c_1} d(v_j, v_i); \quad \acute{U}_{B_i} = \frac{1}{c_2}\sum_{k=1}^{c_2} d(v_k, v_i)$$

where

$$\forall v_j \in \Omega_A; \ \forall v_k \in \Omega_B; \quad \text{and} \ \ c_1 + c_2 = c.$$

The value of γ index is the minimum of the differences between average dissimilarity values of two different class bio-basis strings. A good bio-basis string selection procedure should make the asymmetric dissimilarity between two different class bio-basis strings as high as possible. Actually, the γ index measures how much two different class bio-basis strings are separated. The γ index increases with the increase of separation between the bio-basis strings of two different classes.

3.6 Experimental Results

The performance of the nBBF is compared with that of the existing BBF, while the performance of the new FRDissimilarity+DOR-based bio-basis string selection method (using the concepts of asymmetric dissimilarity-based Fisher ratio, nearest mean classifier, and degree of resemblance) [21] is compared extensively with that of various other related ones. These involve different combinations of the individual components of the hybrid scheme, as well as other related schemes. The algorithms compared are

1. MInformation: the mutual information based method as in (3.10) introduced by Yang and Thomson [41];
2. FRSimilarity: using the similarity-based Fisher ratio as in (3.15) proposed by Berry et al. [8];
3. FRDissimilarity: using the asymmetric dissimilarity-based Fisher ratio as in (3.28); and
4. FRSimilarity+DOR: integration of the FRSimilarity [8] and the concepts of nearest mean classifier and degree of resemblance.

All the algorithms are implemented in C language and run in LINUX environment having machine configuration Pentium IV, 3.2 GHz, 1 MB cache, and 1 GB RAM. To compare the performance of bio-basis string selection methods, α, β, and γ indices are used. The Dayhoff amino acid mutation matrix (Table 3.1) is used to calculate the nongapped pairwise homology score between two subsequences. The performance of two string kernel functions is evaluated based on the prediction accuracy of support vector machine (SVM). The source code of the SVM is obtained from http://svmlight.joachi\discretionary-ms.org/. Details of the SVM and the prediction accuracy are reported next.

3.6.1 Support Vector Machine

The essence in classification is to minimize the probability of error in using the trained classifier, which is refereed to as the structural risk. It has been shown that the SVM [36] is able to minimize the structural risk through finding a unique hyperplane with maximum margin to separate data from two classes. Because of this, the SVM provides best generalization ability on unseen data compared with the other classifiers.

A classification algorithm aims to find a mapping function between input features $\hat{x}$ and a class membership $t \in -1, 1$: $\Gamma = \hat{f}(\hat{x}, w)$, where w is the parameter vector, $\hat{f}(\hat{x}, w)$ the mapping function and Γ the output. With other classification algorithms, the Euclidean distance (error) between Γ and t is minimized to optimize w. This can lead to a biased hyperplane for discrimination.

In search for the best hyperplane, the SVM finds a set of data points that are most difficult training points to classify. These data points are referred to as support vectors. In constructing an SVM classifier, the support vectors are closed in the hyperplane and are located on the boundaries of the margin between two classes. The advantage of using the SVM is that the hyperplane is searched through maximizing the margin. Because of this, the SVM classifier is the most robust, and hence has the best generalization ability. The trained SVM classifier is a linear combination of the similarity between an input and the support vectors. The similarity between an input and the support vectors is quantified by a kernel function defined as: $\psi(\hat{x}, v_i)$, where v_i is the ith support vector. The decision is made using the following equation:

$$\Gamma = \text{sign} \sum a_i t_i \psi(\hat{x}, v_i) \tag{3.44}$$

where t_i is the class label of the ith support vector and a_i is the positive parameter of the ith support vector determined by an SVM algorithm.

However, the kernel function must be specially designed to deal with a data set having nonnumerical attributes such as protein or DNA sequences. In this chapter, a novel string kernel function, termed as the nBBF, is used in the SVM for handling biological subsequences. The performance of the new and the existing string kernel functions is compared by measuring the classification accuracy of the SVM. Four parameters used for this comparison are total accuracy, sensitivity, true positive fraction (TP_f), and true negative fraction (TN_f) as follows:

$$\text{Total Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}; \tag{3.45}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FP}}; \tag{3.46}$$

$$\text{TP}_\text{f} = \frac{\text{TP}}{\text{TP} + \text{FN}}; \qquad \text{TN}_\text{f} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{3.47}$$

Table 3.2 Five Whole HIV Protein Sequences from the NCBI

Accession no	Length	Cleavage sites at P_1
AAC82593	500	132(MA/CA), 363(CA/p2), 377(p2/NC), 432(NC/p1), 448(p1/p6)
AAG42635	498	132(MA/CA), 363(CA/p2), 376(p2/NC), 430(NC/p1), 446(p1/p6)
AAO40777	500	132(MA/CA), 363(CA/p2), 377(p2/NC), 432(NC/p1), 448(p1/p6)
NP_057849	1435	488(TF/PR), 587(PR/RT), 1027(RT/RH), 1147(RH/IN)
NP_057850	500	132(MA/CA), 363(CA/p2), 377(p2/NC), 432(NC/p1), 448(p1/p6)

where TP is the number of true positive predictions, TN is the number of true negative predictions, FP is the number of false positive predictions, and FN is the number of false negative predictions. Hence, the values of TP_f and TN_f represent the precision measures of the positive class and negative class, respectively. To compute the classification accuracy, sensitivity, TP_f, and TN_f of the SVM, the leave-one-out cross-validation (LOOCV) is performed on each data set.

3.6.2 Description of Data Set

To analyze the performance of two string kernel functions and different bio-basis string selection methods, the five whole HIV protein sequences, Cai-Chou HIV data set [9], and caspase cleavage protein sequences are used.

3.6.2.1 Five Whole HIV Protein Sequences

The HIV protease belongs to the family of aspartyl proteases, which have been well-characterized as proteolytic enzymes. The catalytic component is composed of carboxyl groups from two aspartyl residues located in both NH_2- and COOH-terminal halves of the enzyme molecule in the HIV protease [27]. They are strongly substrate-selective and cleavage-specific demonstrating their capability of cleaving large, virus-specific polypeptides called polyproteins between a specific pair of amino acids. Miller et al. showed that the cleavage sites in the HIV polyprotein can extend to an octapeptide region [24]. The amino acid residues within this octapeptide region are represented by P_4-P_3-P_2-P_1-$P_{1'}$-$P_{2'}$-$P_{3'}$-$P_{4'}$, where P_4-P_3-P_2-P_1 is the NH_2-terminal half and $P_{1'}$-$P_{2'}$-$P_{3'}$-$P_{4'}$ the COOH-terminal half. Their counterparts in the HIV protease are represented by S_4-S_3-S_2-S_1-$S_{1'}$-$S_{2'}$-$S_{3'}$-$S_{4'}$ [11]. The HIV protease cleavage site is exactly between P_1 and $P_{1'}$.

The five whole HIV protein sequences have been downloaded from the NCBI (National Center for Biotechnology Information, www.ncbi.nlm.nih.gov). The accession numbers are AAC82593, AAG42635, AAO40777, NP_057849, and NP_057850. Details of these five sequences are included in Table 3.2. Note that MA, CA, NC, TF, PR, RT, RH, and IN are matrix protein, capsid protein, nucleocapsid core protein, transframe peptide, protease, reverse transcriptase, RNAse, and integrase, respectively. They are all cleavage products of the HIV protease. p1, p2, and p6 are also cleavage products [12]. For instance, 132 (MA/CA) means that the cleavage site is between the residues 132 (P_1) and 133 ($P_{1'}$) and the cleavage split the polyprotein producing two functional proteins, the matrix protein and the capsid protein. The subsequences from each of five whole protein sequences are obtained through moving a sliding window with 8 residues. Once a subsequence is produced, it is considered as functional if there is a cleavage site between P_1-$P_{1'}$, otherwise it is labeled as nonfunctional. The total number of subsequences with 8 residues in AAC82593, AAG42635, AAO40777, NP_057849, and NP_057850 are 493, 491, 493, 1428, and 493, respectively.

3.6.2.2 Cai-Chou HIV Data Set

In [9], Cai and Chou have described a benchmark data set of the HIV. It consists of 114 positive oligopeptides and 248 negative oligopeptides, in total 362 subsequences with 8 residues. The data set has been collected from University of Exeter, United Kingdom.

3.6.2.3 Caspase Cleavage Data Set

The programmed cell death, also known as apoptosis, is a gene-directed mechanism, which serves to regulate and control both cell death and tissue homeostasis during the development and the maturation of cells. The importance of apoptosis study is that many diseases such as cancer and ischemic damage result from apoptosis malfunction. A family of cysteine proteases called caspases, which are expressed initially in the cell as proenzymes, is the key to apoptosis [29]. As caspase cleavage is the key to programmed cell death, the study of caspase inhibitors could represent effective drugs against some disease where blocking apoptosis is desirable. Without a careful study of caspase cleavage specificity effective drug design could be difficult.

The 13 protein sequences containing various experimentally determined caspase cleavage sites have been downloaded from the NCBI (www.ncbi.nih.gov). Table 3.3 represents the information of these sequences. Ci depicts the ith caspase. The total number of noncleaved subsequences is about 8,340, while the number of cleaved subsequences is only 18. In total, there are 8,358 subsequences with 8 residues.

Table 3.3 Thirteen caspase cleavage proteins from the NCBI

Proteins	Gene	Length	Cleavage sites
O00273	DFFA	331	117(C3), 224(C3)
Q07817	BCL2L1	233	61(C1)
P11862	GAS2	314	279(C1)
P08592	APP	770	672(C6)
P05067	APP	770	672(C6), 739(C3/C6/C8/C9)
Q9JJV8	BCL2	236	64(C3 and C9)
P10415	BCL2	239	34(C3)
O43903	GAS2	313	278(C)
Q12772	SREBF2	1141	468(C3 and C7)
Q13546	RIPK1	671	324(C8)
Q08378	GOLGA3	1498	59(C2), 139(C3), 311(C7)
O60216	RAD21	631	279(C3/C7)
O95155	UBE4B	1302	109(C3/C7), 123(C6)

3.6.3 Illustrative Example

Consider the data set AAG42635 with sequence length 498. The number of subsequences obtained through moving a sliding window with 8 residues is 491. The parameters used as well as generated in the FRDissimilarity+DOR-based method for selection of bio-basis strings are shown in Table 3.4 only for AAG42635 data, as an example.

In the FRDissimilarity+DOR-based method, the discriminant capability of each subsequence in original set is calculated in terms of the Fisher ratio as in (3.28). The Fisher ratio of each subsequence in original set and reduced set is shown in Fig. 3.1 for this method. While Fig. 3.1a represents the Fisher ratio of the subsequences in original set, Fig. 3.1c shows the same in reduced set considering the values of $\delta = 0.135$ and $\xi = 0.75$. The value of δ is calculated using (3.36). The 86 subsequences present in the reduced set are considered as the bio-basis strings in the FRDissimilarity+DOR-based method. For the purpose of comparison, the 86 subsequences with maximum discriminant capability are selected from the original 491 subsequences of Fig. 3.1a considering the values of $\delta = 0.00$ and $\xi = 1.00$. These subsequences are the bio-basis strings for the FRDissimilarity-based method and shown in Fig. 3.1b. Similarly, the 86 bio-basis strings are selected using the methods proposed by Berry et al. (FRSimilarity) [8] and Yang and Thomson (MInformation) [41], and the FRSimilarity+DOR-based method. In the FRSimilarity [8] based method, the Fisher ratio of each subsequence in original set is calculated using (3.15). Figure 3.2a represents the Fisher ratio of the subsequences in original set, while the bio-basis strings corresponding to the FRSimilarity-based method are shown in Fig. 3.2b. In this case, the 86 subsequences are selected from the original set of Fig. 3.2a as the bio-basis strings which possess strongest discriminant capability. Figure 3.2c represents the results related to the FRSimilarity+DOR, that is, if the

Table 3.4 Comparative analysis of different methods for AAG42635

Sequence length = 498; Number of subsequences, $n = 491$
Value of $\xi = 0.75$; Value of $\delta = 0.135$
Number of bio-basis strings, $c = 86$

Quantitative measures
MInformation [41]: $\alpha = 67.03$; $\beta = 317.66$; $\gamma = 48.49$
FRSimilarity [8]: $\alpha = 68.85$; $\beta = 314.00$; $\gamma = 46.37$
FRDissimilarity: $\alpha = 67.39$; $\beta = 314.00$; $\gamma = 46.42$
FRSimilarity+DOR: $\alpha = 60.81$; $\beta = 350.00$; $\gamma = 45.60$
FRDissimilarity+DOR: $\alpha = 54.11$; $\beta = 366.00$; $\gamma = 57.33$

Prediction accuracy assessment
BBF/MInformation: Accuracy = 0.83; Sensitivity = 0.74; $TP_f = 0.95$; $TN_f = 0.81$
BBF/FRSimilarity: Accuracy = 0.81; Sensitivity = 0.76; $TP_f = 0.94$; $TN_f = 0.85$
BBF/FRDissimilarity: Accuracy = 0.81; Sensitivity = 0.77; $TP_f = 0.95$; $TN_f = 0.85$
BBF/FRSimilarity+DOR: Accuracy = 0.83; Sensitivity = 0.76; $TP_f = 0.95$; $TN_f = 0.86$
BBF/FRDissimilarity+DOR: Accuracy = 0.85; Sensitivity = 0.80; $TP_f = 1.00$; $TN_f = 0.91$

nBBF/MInformation: Accuracy = 0.84; Sensitivity = 0.76; $TP_f = 0.97$; $TN_f = 0.87$
nBBF/FRSimilarity: Accuracy = 0.84; Sensitivity = 0.78; $TP_f = 0.97$; $TN_f = 0.92$
nBBF/FRDissimilarity: Accuracy = 0.84; Sensitivity = 0.79; $TP_f = 0.97$; $TN_f = 0.93$
nBBF/FRSimilarity+DOR: Accuracy = 0.85; Sensitivity = 0.81; $TP_f = 0.97$; $TN_f = 0.93$
nBBF/FRDissimilarity+DOR: Accuracy = 0.92; Sensitivity = 0.87; $TP_f = 1.00$; $TN_f = 0.96$

concepts of nearest mean classifier and degree of resemblance are incorporated in the existing method FRSimilarity [8]. In this case, the value of $\delta = 0.035$ computed using (3.36) and $\xi = 0.75$.

The performance of the nBBF and the FRDissimilarity+DOR-based method for AAG42635 data is shown in Table 3.4, along with the results obtained using the BBF, the existing FRSimilarity [8] and MInformation [41] based bio-basis string selection methods, and the FRDissimilarity and FRSimilarity+DOR-based methods. All the results reported in Table 3.4 establish the superiority of the nBBF and FRDissimilarity+DOR-based bio-basis string selection method over the existing BBF and the bio-basis string selection methods reported in [8, 41] in terms of the α, β, and γ indices as well as the accuracy, sensitivity, TP_f, and TN_f of the SVM.

3.6.4 Performance of Different String Selection Methods

The performance of the FRDissimilarity+DOR-based bio-basis string selection method, along with a comparison with related algorithms, is reported in Tables 3.5,

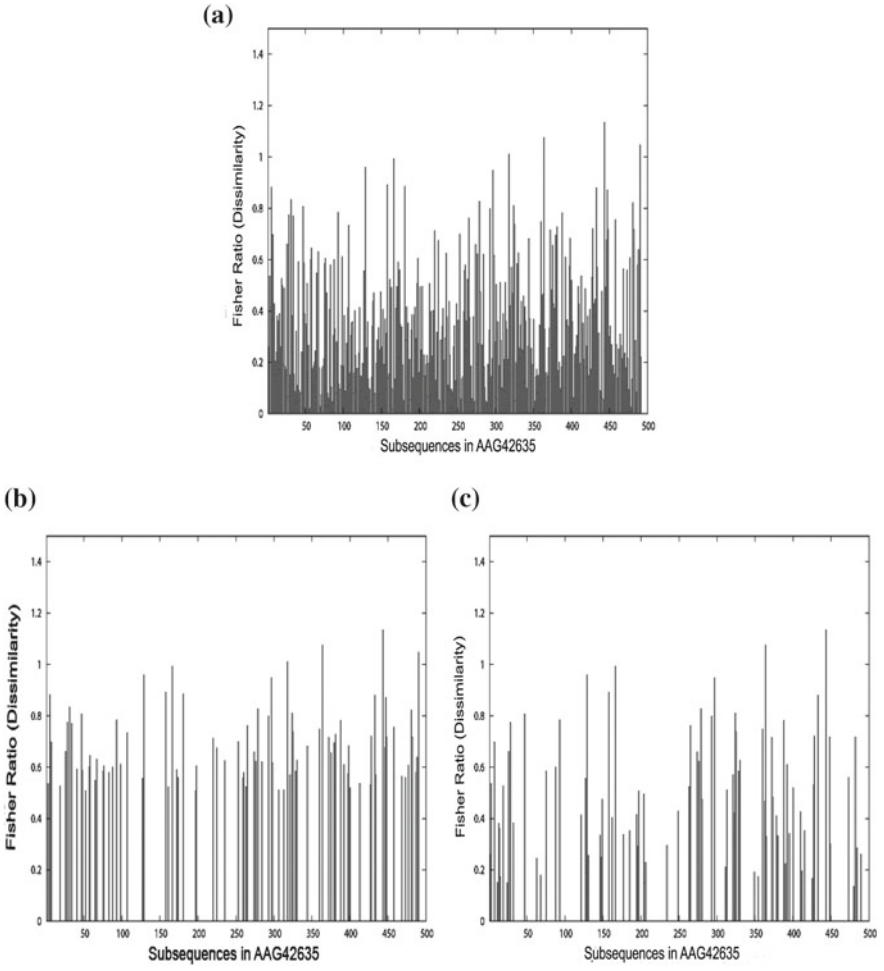

Fig. 3.1 Discriminant capability of subsequences in FRDissimilarity and FRDissimilarity+DOR based methods. **a** Original data set. **b** Reduced set for $\delta = 0.000$; $\xi = 1.00$. **c** Reduced set for $\delta = 0.135$; $\xi = 0.75$

3.6, 3.7, 3.8. Subsequent discussions analyze the results presented in these tables with respect to α, β, γ, and execution time.

The LOOCV is performed on each data set. The means and standard deviations of the α, β, and γ indices are computed for all data sets. Tests of significance are performed for the inequality of means (of the α, β, or γ) obtained using the new bio-basis string selection method and the other related algorithms compared. Since both mean pairs and the variance pairs are unknown and different, a generalized version of t-test is used here. The above problem is the classical Behrens-Fisher problem in hypothesis testing. The test statistic, which is described and tabled in [5], is of the

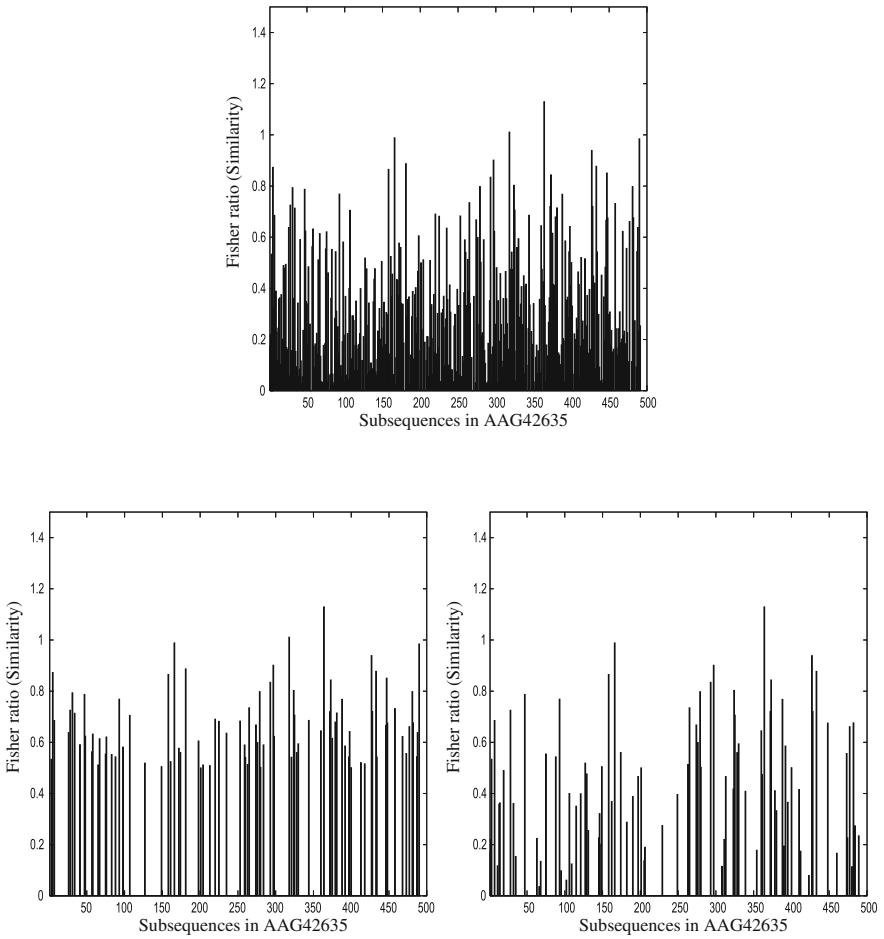

Fig. 3.2 Discriminant capability of subsequences in FRSimilarity [8] and FRSimilarity+DOR-based methods. **a** Original data set. **b** Reduced set for $\delta = 0.000$; $\xi = 1.00$. **c** Reduced set for $\delta = 0.035$; $\xi = 0.75$

form

$$\hat{t} = \frac{\mu_1 - \mu_2}{\sqrt{\lambda_1 {\sigma_1}^2 + \lambda_2 {\sigma_2}^2}} \tag{3.48}$$

where μ_1, μ_2 are the means, σ_1, σ_2 the standard deviations, and $\lambda_1 = 1/n_1$, $\lambda_2 = 1/n_2$, n_1, n_2 are the number of observations. Tables 3.6, 3.7 and 3.8 report the individual means and standard deviations, and the value of test statistic computed and the corresponding tabled values at an error probability level of 0.001. Figures in parentheses indicate the computed value of test statistic and tabled value,

Table 3.5 Performance of FRDissimilarity+DOR based method for different values of ξ

Value of		Description of Different Data Sets						
ξ	Index	AAC82593	AAG42635	AAO40777	NP_057849	NP_057850	Cai-Chou	Caspase cleavage
0.60	c	14	11	16	22	14	14	37
	α	105.96	86.96	113.53	122.51	105.97	110.63	105.56
	β	352.00	310.00	352.00	350.00	352.00	292.00	380.00
	γ	193.54	92.00	193.00	119.80	193.54	84.26	155.56
0.65	c	34	28	33	56	34	32	75
	α	93.68	82.60	93.14	95.98	93.68	91.95	93.20
	β	332.00	398.00	324.00	384.00	332.00	332.00	346.00
	γ	80.86	89.00	79.59	110.96	80.86	69.28	142.86
0.70	c	58	42	59	101	58	36	166
	α	72.52	66.23	70.03	77.42	72.53	98.50	76.11
	β	350.00	354.00	350.00	362.00	350.00	326.00	354.00
	γ	70.51	74.66	67.67	127.95	70.518	68.79	137.45
0.75	c	88	86	93	211	88	58	393
	α	52.23	54.11	52.61	51.61	53.23	67.89	46.20
	β	350.00	366.00	360.00	370.00	350.00	320.00	376.10
	γ	61.57	57.33	62.80	128.01	61.57	58.68	85.03
0.80	c	97	86	97	278	97	30	863
	α	53.43	54.83	53.43	51.90	53.44	75.49	46.25
	β	318.00	328.00	322.00	322.00	318.00	268.00	376.00
	γ	60.49	56.197	61.49	90.97	60.49	58.66	54.98
0.85	c	40	34	46	146	40	15	1628
	α	59.289	61.76	55.93	55.68	105.97	67.37	33.65
	β	286.00	286.00	322.00	350.00	300.00	266.00	376.00
	γ	69.48	60.96	61.49	119.80	88.61	78.80	48.87

respectively. If the computed value is greater than the tabled value, the means are significantly different.

3.6.4.1 Optimum Value of ξ

The threshold ξ has an influence on the performance of the FRDissimilarity+DOR-based bio-basis string selection method. It controls the degree of redundancy between two subsequences. To find out the optimum value of ξ, extensive experiments are carried out for different values of ξ. It may be noted that the optimum choice of c (the number of bio-basis strings) is a function of ξ. Table 3.5 represents the mean values of α, β, and γ along with the number of bio-basis strings c obtained using the FRDissimilarity+DOR-based method for all the data sets reported in Sect. 3.6.2. Results are reported for different values of ξ. It is seen from the results of Table 3.5 that as the value of ξ increases, the FRDissimilarity+DOR-based method consistently

Table 3.6 Comparative performance analysis of FRDissimilarity and FRDissimilarity+DOR

Data sets	Methods/Algorithms	α Index		β Index		γ Index		Time
		Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	(ms)
AAC	FRDissimilarity+DOR	52.23	5.273	350.00	7.100	61.57	3.177	2732
82593	FRDissimilarity	66.68	10.036	318.00	11.624	52.70	7.033	2378
		(28.30	3.1068)	(52.16	3.1068)	(25.52	3.1068)	
AAG	FRDissimilarity+DOR	54.11	2.301	366.00	6.153	57.33	3.807	2724
42635	FRDissimilarity	67.39	4.802	314.00	10.611	46.42	6.174	2335
		(55.26	3.1069)	(93.94	3.1069)	(33.33	3.1069)	
AAO	FRDissimilarity+DOR	52.61	3.274	360.00	6.041	62.80	2.816	2768
40777	FRDissimilarity	64.86	7.029	348.00	9.642	53.80	9.020	2389
		(35.08	3.1068)	(23.42	3.1068)	(21.15	3.1068)	
NP_O	FRDissimilarity+DOR	51.61	6.317	370.00	5.161	128.01	4.007	7749
57849	FRDissimilarity	67.90	9.186	368.00	9.117	97.00	9.877	7425
		(55.22	3.0959)	(7.21	3.0959)	(109.94	3.0959)	
NP_O	FRDissimilarity+DOR	53.23	5.013	350.00	7.100	61.57	3.161	2778
57850	FRDissimilarity	66.57	9.134	310.80	11.607	52.75	7.013	2356
		(28.43	3.1068)	(63.97	3.1068)	(25.46	3.1068)	
Cai-Chou	FRDissimilarity+DOR	67.89	3.104	320.00	5.106	58.68	3.007	2487
HIV Data	FRDissimilarity	77.89	6.809	300.00	9.174	48.74	9.004	2172
		(25.43	3.1129)	(36.24	3.1129)	(19.92	3.1129)	
Caspase	FRDissimilarity+DOR	46.20	2.361	376.10	7.674	85.03	4.166	48529
Cleavage	FRDissimilarity	58.09	5.116	376.00	9.827	84.00	10.627	43819
		(192.92	3.0912)	(**0.73**	**3.0912**)	(8.25	3.0912)	

achieves better performance. That is, the value of α decreases, and the values of β and γ increase with the increase in ξ. The best performance with respect to α, β, and γ is achieved with $\xi = 0.75$. However, for $\xi > 0.75$, the performance decreases with the increase in ξ. That is, the best performance of the FRDissimilarity+DOR-based bio-basis string selection method is obtained when one subsequence x_j is considered to contain redundant information of another subsequence x_i if the homology score between them is greater than or equal to 75 % of the maximum homology score of x_i. Hence, to achieve best performance using the FRDissimilarity+DOR-based bio-basis string selection method, the subsequence x_j is considered as a redundant one of the subsequence x_i if $h(x_j, x_i) \geq 0.75 \times h(x_i, x_i)$, where $h(x_j, x_i)$ and $h(x_i, x_i)$ represent the homology score between x_j and x_i, and the maximum homology score of x_i, respectively. In other words, the degree of resemblance between two subsequences selected as two nonredundant bio-basis strings must be less than 0.75.

Table 3.7 Comparative performance analysis of biological similarity and dissimilarity

Data sets	Methods/Algorithms	α Index		β Index		γ Index		Time
		Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	(ms)
AAC	FRDissimilarity+DOR	52.23	5.273	350.00	7.100	61.57	3.177	2732
82593	FRSimilarity+DOR	60.00	7.318	340.00	9.034	54.89	6.820	2721
		(19.13	3.1068)	(19.32	3.1068)	(19.71	3.1068)	
AAG	FRDissimilarity+DOR	54.11	2.301	366.00	6.153	57.33	3.807	2724
42635	FRSimilarity+DOR	60.81	3.174	350.00	8.805	45.60	5.924	2711
		(37.87	3.1069)	(33.01	3.1069)	(36.91	3.1069)	
AAO	FRDissimilarity+DOR	52.61	3.274	360.00	6.041	62.80	2.816	2768
40777	FRSimilarity+DOR	60.00	5.945	354.00	8.113	60.00	5.117	2745
		(24.18	3.1068)	(13.17	3.1068)	(10.64	3.1068)	
NP_O	FRDissimilarity+DOR	51.61	6.317	370.00	5.161	128.01	4.007	7749
57849	FRSimilarity+DOR	55.48	7.103	342.00	8.081	98.38	7.821	7689
		(15.38	3.0959)	(110.35	3.0959)	(127.41	3.0959)	
NP_O	FRDissimilarity+DOR	53.23	5.013	350.00	7.100	61.57	3.161	2778
57850	FRSimilarity+DOR	60.20	6.819	341.23	9.016	55.68	7.008	2756
		(18.29	3.1068)	(16.97	3.1068)	(17.01	3.1068)	
Cai-Chou	FRDissimilarity+DOR	67.89	3.104	320.00	5.106	58.68	3.007	2487
HIV Data	FRSimilarity+DOR	60.46	4.713	286.78	6.117	50.67	5.192	2429
		(**−25.05**	**3.1129**)	(79.32	3.1129)	(25.40	3.1129)	
Caspase	FRDissimilarity+DOR	46.20	2.361	376.10	7.674	85.03	4.166	48529
Cleavage	FRSimilarity+DOR	58.09	2.813	290.00	8.130	79.08	7.314	46537
		(295.99	3.0912)	(704.08	3.0912)	(64.62	3.0912)	

3.6.4.2 Degree of Resemblance and Nearest Mean Classifier

The main objective of introducing the concepts of degree of resemblance and nearest mean classifier in the FRDissimilarity+DOR-based bio-basis string selection method is to reduce the number of bio-basis strings from the whole set of subsequences. While the concept of degree of resemblance is introduced to eliminate the redundant subsequences, the principle of nearest mean classifier is used to discard the nonrelevant subsequences [21].

In order to establish the importance of both degree of resemblance and nearest mean classifier, extensive experiments are carried out. Table 3.6 provides comparative results of the bio-basis string selection methods with and without considering the above two concepts. The discriminant capability of each subsequence in both cases (FRDissimilarity+DOR and FRDissimilarity) is calculated using the Fisher ratio as in (3.28), while the value of ξ is set to 0.75 and the value of δ is computed according to (3.36) for the FRDissimilarity+DOR based method. The FRDissimilarity+DOR-based method is found to improve the performance of the FRDissimilarity-based method in terms of α, β, and γ. Regarding statistical significance tests, it can be seen from Table 3.6 that out of 21 comparisons, the FRDissimilarity+DOR-based

Table 3.8 Comparative performance analysis of different methods

Data sets	Methods/Algorithms	α Index		β Index		γ Index		Time (ms)
		Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	
AAC8	FRDissimilarity+DOR	52.23	5.273	350.00	7.100	61.57	3.177	2732
2593	FRSimilarity [8]	67.13	12.149	312.68	13.183	52.70	7.017	2217
		(24.98	3.1068)	(55.34	3.1068)	(25.57	3.1068)	
	MInformation [41]	62.18	6.928	322.60	9.615	52.81	6.801	2561
		(25.38	3.1068)	(50.90	3.1068)	(25.91	3.1068)	
AAG4	FRDissimilarity+DOR	54.11	2.301	366.00	6.153	57.33	3.807	2724
2635	FRSimilarity [8]	68.85	5.003	314.00	13.081	46.37	6.890	2192
		(59.31	3.1069)	(79.71	3.1069)	(30.85	3.1069)	
	MInformation [41]	67.03	3.018	317.66	9.013	48.49	5.800	2568
		(75.44	3.1069)	(98.15	3.1069)	(28.23	3.1069)	
AAO4	FRDissimilarity+DOR	52.61	3.274	360.00	6.041	62.80	2.816	2768
0777	FRSimilarity [8]	65.17	7.813	328.10	11.081	52.33	8.724	2120
		(32.92	3.1068)	(56.12	3.1068)	(25.36	3.1068)	
	MInformation [41]	63.19	7.008	351.30	9.001	54.10	9.007	2412
		(30.37	3.1068)	(17.82	3.1068)	(20.47	3.1068)	
NP_O5	FRDissimilarity+DOR	51.61	6.317	370.00	5.161	128.01	4.007	7749
7849	FRSimilarity [8]	70.12	12.823	317.50	10.629	93.70	11.066	7108
		(28.75	3.0959)	(167.90	3.0959)	(110.16	3.0959)	
	MInformation [41]	64.18	9.672	332.70	8.728	98.30	10.092	7468
		(41.12	3.0959)	(139.01	3.0959)	(103.40	3.0959)	
NP_O5	FRDissimilarity+DOR	53.23	5.013	350.00	7.100	61.57	3.161	2778
7850	FRSimilarity [8]	68.13	11.940	304.84	13.004	52.75	6.898	2071
		(25.55	3.1068)	(67.68	3.1068)	(25.81	3.1068)	
	MInformation [41]	62.75	7.004	337.10	10.012	55.11	6.054	2683
		(24.54	3.1068)	(23.34	3.1068)	(21.00	3.1068)	
Cai-Chou	FRDissimilarity+DOR	67.89	3.104	320.00	5.106	58.68	3.007	2487
HIV Data	FRSimilarity [8]	77.17	7.002	285.90	12.188	48.60	10.361	2026
		(23.05	3.1129)	(49.10	3.1129)	(17.78	3.1129)	
	MInformation [41]	71.63	6.624	306.10	8.103	50.65	9.882	2216
		(9.73	3.1129)	(27.61	3.1129)	(14.79	3.1129)	
Caspase	FRDissimilarity+DOR	46.20	2.361	376.10	7.674	85.03	4.166	48529
Cleavage	FRSimilarity [8]	61.08	8.025	288.50	11.628	78.06	13.160	40173
		(162.62	3.0912)	(574.83	3.0912)	(46.16	3.0912)	
	MInformation [41]	57.10	6.138	328.40	9.713	80.33	5.381	46119
		(151.53	3.0912)	(352.28	3.0912)	(63.14	3.0912)	

method is found to provide significantly better results in 20 comparisons. Only for caspase cleavage data set, the value of β of the FRDissimilarity+DOR-based method is found to be better, but not significantly. The corresponding entry is marked bold in Table 3.6.

The execution time required for the FRDissimilarity+DOR-based method is higher compared to that of the FRDissimilarity-based method. However, the FRDissimilarity+DOR-based method selects the bio-basis strings that are more relevant and distinct or nonredundant. Hence, to improve the performance of the FRDissimilarity-based method by eliminating similar and nonrelevant bio-basis strings, the degree of resemblance and the nearest mean classifier should be incorporated with the FRDissimilarity-based method. In effect, the FRDissimilarity+DOR-based method selects a reduced set of relevant and nonredundant bio-basis strings that helps to generate distinct and useful features in numerical feature space.

3.6.4.3 Importance of Asymmetricity of Biological Dissimilarity

The FRDissimilarity+DOR-based method calculates the relevance or discriminant capability of each subsequence based on the principle of asymmetric biological dissimilarity as in (3.28), while the existing method FRSimilarity considers the biological similarity to compute the discriminant capability as in (3.15). As a result, the FRDissimilarity+DOR-based method takes into account the zero-mean, higher order (upto fourth order) moment of similarity as well as the maximum homology score of the subsequence, while the existing FRSimilarity-based method does not consider the maximum homology score and considers only the zero-mean, first- and second-order moment of similarity.

In order to establish the importance of biological dissimilarity over biological similarity, extensive experiments are carried out. Table 3.7 provides comparative results of the FRDissimilarity+DOR and FRSimilarity+DOR-based bio-basis string selection methods considering $\xi = 0.75$. The dissimilarity based method is found to improve the performance in terms of α, β, and γ with comparable time. Statistical significance tests are also presented for all the comparisons, and in 20 of 21 comparisons, the FRDissimilarity+DOR-based method performs significantly better than the FRSimilarity+DOR based method. Hence, the FRDissimilarity+DOR based method selects the relevant and distinct bio-basis strings more accurately compared to the FRSimilarity+DOR-based method. Only, in case of Cai-Chou HIV data set, the value of α is significantly better in similarity-based approach, while the values of β and γ are higher in dissimilarity-based approach. This case is denoted by bolded entry in Table 3.7.

3.6.4.4 Comparative Analysis of Different Methods

Finally, Table 3.8 provides the comparative results of the FRDissimilarity+DOR and existing (FRSimilarity and MInformation) algorithms for the protein sequences reported in Sect. 3.6.2. It is seen that the FRDissimilarity+DOR-based method produces bio-basis strings having the lowest α value and highest β and γ values for all the cases. Table 3.8 also reports the execution time (in milli second) of different algorithms for all protein data sets. The execution time required for the

FRDissimilarity+DOR-based method is higher compared to that of the existing methods. For the FRSimilarity, although the execution time is less, the performance is significantly poorer than that of the MInformation and FRDissimilarity+DOR. From the statistical significance tests presented in Table 3.8, it can be seen that in all 42 comparisons, the FRDissimilarity+DOR-based bio-basis string selection method is found to provide significantly better results compared to existing methods.

3.6.5 Performance of Novel Bio-Basis Function

The performance of the nBBF (novel bio-basis function) is presented with respect to the prediction accuracy of the SVM (support vector machine). The nBBF normalizes the asymmetric dissimilarity between a bio-basis string and a subsequence using the zone of influence or variance of that bio-basis string as in (3.20), rather than using the maximum homology score of that bio-basis string as in the existing BBF [8, 41].

Table 3.9 provides the comparative results of these two kernel functions for the protein sequences reported in Sect. 3.6.2. The bio-basis strings for each data sets are generated using the FRDissimilarity+DOR-based method. The zone of influence of each bio-basis string is calculated using (3.21) for the nBBF. The LOOCV is performed to compute the prediction accuracy of the SVM. From the results reported in Table 3.9 with respect to TN_f, TP_f, sensitivity, and accuracy, it is seen that the nBBF provides better results for all protein data sets. That is, the nBBF transforms nonnumerical sequence space to numerical feature space more accurately than the existing BBF. All the results reported in Table 3.9 establish that the concept of zone of influence introduced in the nBBF efficiently normalizes the asymmetric dissimilarity taking into account the influence (impact) of each bio-basis string in nonnumerical sequence space.

The following conclusions can be drawn from the results reported in Tables 3.5, 3.6, 3.7, 3.8, 3.9:

1. It is seen that the FRDissimilarity+DOR-based bio-basis string selection method is superior to the existing FRSimilarity [8] and MInformation [41] based methods. However, the FRSimilarity and MInformation methods require slightly lesser time compared to that of the FRDissimilarity+DOR-based method. But, the performance of existing methods is significantly poorer than the new method.
2. The FRDissimilarity+DOR-based method is found to improve the performance of the existing methods (in terms of α, β, and γ) significantly.

The best performance of the FRDissimilarity+DOR-based method in terms of α, β, and γ is achieved due to the following reasons:

1. The asymmetric biological dissimilarity is more effective compared to the symmetric similarity measure to calculate the discriminant capability or relevance of the subsequences in terms of the Fisher ratio.

Table 3.9 Comparative performance of two kernel functions

Data Sets	Functions	TN_f	TP_f	Sensitivity	Accuracy
AAC8	nBBF	0.99	1.00	0.95	0.93
2593	BBF	0.97	1.00	0.88	0.86
AAG4	nBBF	0.96	1.00	0.87	0.92
2635	BBF	0.91	1.00	0.80	0.85
AAO4	nBBF	0.96	0.99	0.91	0.92
0777	BBF	0.92	0.99	0.85	0.86
NP_05	nBBF	0.97	0.94	0.91	0.89
7849	BBF	0.88	0.93	0.90	0.83
NP_O5	nBBF	0.99	1.00	0.91	0.91
7850	BBF	0.98	1.00	0.88	0.90
Cai-Chou	nBBF	0.94	0.96	0.91	0.90
HIV Data	BBF	0.87	0.90	0.76	0.88
Caspase	nBBF	0.95	0.90	0.89	0.92
Cleavage	BBF	0.94	0.91	0.86	0.91

2. The principle of nearest mean classifier and the concept of degree of resemblance enable efficient selection of relevant and distinct bio-basis strings. As a result, it reduces the nonrelevant and redundant features in numerical feature space.

In effect, a reduced set of most relevant and nonredundant bio-basis strings is obtained using the FRDissimilarity+DOR-based bio-basis string selection method. The concept of zone of influence introduced in novel bio-basis function (nBBF) normalizes the biological dissimilarity. As it takes into account the influence of each bio-basis string in nonnumerical sequence space, the nBBF transforms nonnumerical sequence space to numerical feature space more accurately. Hence, the best performance of the nBBF in terms of the prediction accuracy of the SVM is achieved.

3.7 Conclusion and Discussion

The major contribution of this chapter is to present a novel string kernel function based on the principle of asymmetricity of dissimilarity and the concept of zone of influence of bio-basis string. An efficient method is reported for selection of a reduced set of most relevant and nonredundant bio-basis strings. Some new measures based on homology score are also presented to evaluate the quality of selected bio-basis strings. Moreover, the current chapter demonstrates the effectiveness of the new string kernel function and the bio-basis string selection method, along with a comparison with existing string kernel function and related bio-basis string selection methods, on different protein data sets.

Some of the indices (for example, α, β, and γ) used for evaluating the quality of selected bio-basis strings may be used in a suitable combination to act as the

objective function of an evolutionary algorithm, for generating a reduced set of most relevant bio-basis strings. This formulation is geared toward maximizing the utility of the biological content with respect to bio-basis string selection task.

So far we have described in Chap. 2 and in this chapter different classification methodologies with extensive experimental results demonstrating their characteristic features. The next four chapters deal with different feature selection approaches, along with some of the specific real-life problems in computational biology and bioinformatics, namely, selection of effective molecular descriptors to predict biological activity of molecules, selection of discriminative genes from high dimensional microarray data, identification of disease-related genes, and selection of discriminative microRNAs from expression data.

References

1. Aho AV, Corasick M (1975) Efficient string matching: an aid to bibliographic search. Commun ACM 18(6):333–340
2. Altschul SF, Boguski MS, Gish W, Wootton JC (1994) Issues in searching molecular sequence databases. Nat Genet 6(2):119–129
3. Altschul SF, Gish W, Miller W, Myers E, Lipman DJ (1990) Basic local alignment search tool. J Mol Biol 215:403–410
4. Arrigo P, Giuliano F, Damiani G (1991) Identification of a new Motif on nucleic acid sequence data using Kohonen's self-organising map. Comput Appl Biosci 7(3):353–357
5. Aspin A (1949) Tables for use in comparisons whose accuracy involves two variances separately estimated. Biometrika 36(3–4):290–296
6. Baldi P, Brunak S (1998) Bioinformatics: the machine learning approach. MIT Press, Cambridge
7. Baldi P, Pollastri G, Anderson CA, Brunak S (1995) Matching protein Beta-sheet partners by feedforward and recurrent neural networks. Proc Int Conf Intell Syst Mol Biol 8:25–36
8. Berry EA, Dalby AR, Yang ZR (2004) Reduced bio-basis function neural network for identification of protein phosphorylation sites: comparison with pattern recognition algorithms. Comput Biol Chem 28(1):75–85
9. Cai YD, Chou KC (1998) Artificial neural network model for predicting HIV protease cleavage sites in protein. Adv Eng Softw 29(2):119–128
10. Cai YD, Liu XJ, Xu XB, Chou KC (2002) Support vector machines for predicting the specificity of GalNAc-transferase. Peptides 23:205–208
11. Chou KC (1993) A vectorised sequence-coupling model for predicting HIV protease cleavage sites in proteins. J Biol Chem 268(23):16, 938–16, 948
12. Chou KC (1996) Prediction of human immunodeficiency virus protease cleavage sites in proteins. Anal Biochem 233(1):1–14
13. Dayhoff MO, Schwartz RM, Orcutt BC (1978) A model of evolutionary change in proteins. Matrices for detecting distant relationships. Atlas Protein Seq Struct 5:345–358
14. Duda RO, Hart PE, Stork DG (1999) Pattern classification and scene analysis. Wiley, New York
15. Henikoff S, Henikoff JG (1992) Amino acid substitution matrices from protein blocks. In: Proc Nat Acad Sci USA 89:10, 915–10, 91
16. Itoh M, Goto S, Akutsu T, Kanehisa M (2005) Fast and accurate database homology search using upper bounds of local alignment scores. Bioinformatics 21(7):912–921
17. Johnson MS, Overington JP (1993) A structural basis for sequence comparisons: an evaluation of scoring methodologies. J Mol Biol 233(4):716–738

18. Lui YM, Cheng HD (1996) A new peak selection criterion based on minimizing the classification error. Inf Sci 94(1–4):213–233
19. Maji P, Pal SK (2007) Protein sequence analysis using relational soft clustering algorithms. Int J Comput Math 84(5):599–617
20. Maji P, Pal SK (2007) Rough-Fuzzy C-medoids algorithm and selection of bio-basis for amino acid sequence analysis. IEEE Trans Knowl Data Eng 19(6):859–872
21. Maji P, Das C (2010) Efficient design of bio-basis function to predict protein functional sites using Kernel-based classifiers. IEEE Trans NanoBiosci 9(4):242–249
22. Maji P, Das C (2010) Protein functional sites prediction using modified bio-basis function and quantitative indices. IEEE Trans NanoBiosci 9(4):250–257
23. Maji P, Pal SK (2012) Rough-fuzzy pattern recognition: applications in bioinformatics and medical imaging. Wiley-IEEE Computer Society Press, New Jersey
24. Miller M, Schneider J, Sathayanarayana BK, Toth MV, Marshall GR, Clawson L, Selk L, Kent SBH, Wlodawer A (1989) Structure of complex of synthetic HIV-1 protease with substrate-based inhibitor at 2.3 a resolution. Science 246(4934):1149–1152
25. Minakuchi Y, Satou K, Konagaya A (2002) Prediction of protein-protein interaction sites using support vector machines. Genome Inform 13:322–323
26. Narayanan A, Wu XK, Yang ZR (2002) Mining viral protease data to extract cleavage knowledge. Bioinformatics 18:5–13
27. Pearl LH, Taylor WR (1987) A structural model for the retroviral proteases. Nature 329(6137):351–354
28. Qian N, Sejnowski TJ (1988) Predicting the secondary structure of globular proteins using neural network models. J Mol Biol 202(4):865–884
29. Rohn TT, Cusack SM, Kessinger SR, Oxford JT (2004) Caspase activation independent of cell death is required for proper cell dispersal and correct morphology in PC12 cells. Exp Cell Res 295(1):215–225
30. Searls DB (1996) Sequence alignment through pictures. Trends Genet 12:35–37
31. Searls DB, Murphy KP (1995) Automata-theoretic models of mutation and alignment. In: Proceedings of the 3rd international conference on intelligent systems for molecular biology, The AAAI Press, pp 341–349
32. Shannon C, Weaver W (1964) The mathematical theory of communication. University of Illinois Press, Champaign
33. Stojmirovic A (2004) Quasi-metric spaces with measure. Topol Proc 28(2):655–671
34. Thompson K (1968) Regular expression search algorithm. Commun ACM 11(6):419–422
35. Thomson R, Hodgman C, Yang ZR, Doyle AK (2003) Characterising Proteolytic cleavage site activity using bio-basis function neural network. Bioinformatics 19(14):1741–1747
36. Vapnik V (1995) The nature of statistical learning theory. Springer-Verlag, New York
37. Yang ZR (2004) Biological application of support vector machines. Briefings Bioinform 5(4):328–338
38. Yang ZR (2005) Orthogonal Kernel machine for the prediction of functional sites in proteins. IEEE Trans Syst Man Cybern Part B Cybern 35(1):100–106
39. Yang ZR (2005) Prediction of caspase cleavage sites using bayesian bio-basis function neural networks. Bioinformatics 21(9):1831–1837
40. Yang ZR, Chou KC (2004) Predicting the O-Linkage sites in glycoproteins using bio-basis function neural networks. Bioinformatics 20(6):903–908
41. Yang ZR, Thomson R (2005) Bio-basis function neural network for prediction of protease cleavage sites in proteins. IEEE Trans Neural Netw 16(1):263–274
42. Yang ZR, Thomson R, McNeil P, Esnouf R (2005) RONN: use of the bio-basis function neural network technique for the detection of natively disordered regions in proteins. Bioinformatics 21(16):3369–3376

Part II
Feature Selection

Chapter 4
Rough Sets for Selection of Molecular Descriptors to Predict Biological Activity of Molecules

4.1 Introduction

In conventional drug design, the drug discovery proceeds largely by trial and error synthesizing thousands of molecules. Although this approach is the most effective method to discover drugs, it is very financially expensive and labor intensive. The conventional drug design method is improved by a nonconventional method, termed as computer-aided drug design (CADD) [2]. The CADD helps in predicting biological activity of a hypothetical molecule and guides scientists toward a specific direction to develop a drug by predicting a molecule with effective biological activity or molecular property against a target molecule. In effect, it minimizes both time and cost. Two well-known approaches are generally taken for prediction: structure-based method and quantitative structure activity relationship (QSAR) method [29]. In structure-based method, the procedure starts with the known three-dimensional structure of a target molecule, where the goal is to design a ligand or drug that can enhance or decrease the activity of the target molecule. Whereas the QSAR method predicts the activity of hypothetical compounds based on the assayed activity of previously synthesized one [13].

The QSAR is the process by which chemical structure is quantitatively correlated with a well-defined process such as biological activity or other molecular property. Biological activity can be expressed quantitatively as in the concentration of a substance required to give a certain biological response. Additionally, when physiochemical properties or structures are expressed by numbers, one can form a mathematical relationship or quantitative structure activity relationship between the two. The mathematical expression can then be used to predict the biological response of other unknown chemical structures. The properties that describe the molecule quantitatively are known as molecular descriptors. Molecular descriptors can be obtained by calculated methods or experimental methods. In calculated method, a mathematical procedure is used that transforms chemical information into a number such as surface areas (polar, non-polar), dipole moment, and volume. On the other hand, in experimental method, some standardized experiments are conducted to measure

P. Maji and S. Paul, *Scalable Pattern Recognition Algorithms*,
DOI: 10.1007/978-3-319-05630-2_4,

a molecular descriptor such as melting point, partition coefficients, and refractive index. The molecular descriptors describe different aspects of a molecule, compare different molecular structures, different conformations of same molecule, and database storage, and relate structure to activity [24, 29, 31, 54].

Many approaches have been proposed to generate a error free method for predicting biological activity or other chemical property of a molecule. Ozdemir et al. [40] used genetic algorithm to select a subset of molecular descriptors and the significance of these descriptors has been evaluated by a multilayer perceptron. Guha and Jurs [13, 14] used correlation, simulated annealing, and genetic algorithm to obtain the best subset of descriptors. Both linear and nonlinear predictive models have been used to establish the significance of selected descriptors. Similar type of work has been done by Leardi and Gonzalez [30], where genetic algorithm has been used for descriptor selection and partial least square method for prediction. Sventik et al. [53] used the concept of ensemble method for compound classification and biological activity prediction. In [19], Jain et al. have used steric and polar descriptors to predict the biological activity. Tuppurainen et al. [56] and Turner et al. [57] have used an electronic eigenvalue molecular descriptor and a molecular vibration based descriptor, respectively, to relate structure and activity of steroid data. Different three-dimensional molecular descriptors have been proposed in [5, 47] to forecast the biological activity. A different approach based on fuzzy regression has been used to predict the biological activity of persistent organic pollutants [58]. Kumar et al. [28] used a method based on fuzzy mappings for the QSAR modeling, while a neural and neuro-fuzzy model have been used in [39] for prediction of toxic action of phenols. On the other hand, Zhou et al. [64] have developed a robust boosting partial least square method for modeling the antagonisms of angiotensin II antagonists.

However, among the large amount of descriptors, only a small fraction is effective for performing the predictive modeling task. Also, a small subset of descriptors is desirable in developing QSAR data based predicting tools for delivering precise, reliable, and interpretable results. With the descriptor selection results, the cost of biological experiment and decision can be greatly reduced by analyzing only the effective descriptors. Hence, identifying a reduced set of most relevant descriptors is the goal of descriptor selection. The small number of molecules and a large number of descriptors make this problem a more relevant and challenging problem in the QSAR method. This is an important problem in pattern recognition and machine learning and referred to as feature selection [10].

Feature selection or dimensionality reduction of a data set is an essential preprocessing step used for pattern recognition, data mining, and machine learning. It is an important problem related to mining large data sets, both in dimension and size. Prior to analysis of the data set, preprocessing the data to obtain a smaller set of representative features and retaining the optimal salient characteristics of the data not only decrease the processing time but also lead to more compactness of the models learned and better generalization. Hence, the general criterion for reducing the dimension is to preserve most relevant information of the original data according to some optimality criteria.

Conventional methods of feature selection involve evaluating different feature subsets using some index and selecting the best among them. Depending on the way of computing the feature evaluation index, feature selection methods are generally divided into two broad categories, namely, filter approach [10, 26] and wrapper approach [10, 25]. In filter approach, the algorithms do not perform classification of the data in the process of feature evaluation. Before application of the actual learning algorithm, the best subset of features is selected in one pass by evaluating some predefined criteria, which are independent of the actual generalization performance of the learning machine. Hence, the filter approach is computationally less expensive and more general [10, 26].

On the other hand, in its most general formulation, the wrapper approach consists of using the prediction performance of a given learning machine to assess the relative usefulness of different subsets of features. Since the wrapper approach uses the learning machine as a black box, it generally outperforms the filter approach in the aspect of final predictive accuracy of the learning machine. However, it is computationally more expensive than that of the filter approach [10, 25]. An efficient but less universal version of the wrapper approach is the embedded method, which performs feature selection in the process of training and is usually specific to given learning machine. However, the embedded approach is much intricate and limited to a specific learning machine [15, 44–46].

Rough set theory is a new paradigm to deal with uncertainty, vagueness, and incompleteness. It is proposed for indiscernibility in classification according to some similarity. The rough set theory has been applied successfully to feature selection of discrete valued data [42, 49, 51]. Given a data set with discretized attribute values, it is possible to find a subset of the original attributes using rough set theory that are the most informative; all other attributes can be removed from the data set with minimal information loss. From the dimensionality reduction perspective, informative features are those that are most useful in determining classifications from their values [7, 62].

One of the popular rough set-based feature selection algorithms is quick reduct algorithm [8, 9] in which the dependency or quality of approximation of single attribute is first calculated with respect to the class labels or decision attribute. After selecting the best attribute, other attributes are added to it to produce better quality. Additions of attributes are stopped when the final subset of attributes has the same quality as that of maximum possible quality of the data set or the quality of the selected attributes remains same.

A reduct with effective attributes can also be obtained from the discernibility matrix-based method [27, 50]. The matrix is developed by considering those attributes which differentiate objects. A discernibility function can then be defined for discernibility matrix data. This function generates all the minimal reducts. However, this approach is computationally very costly. On the other hand, the variable precision rough set-based attribute selection algorithm [65] is an important method with better generalization ability to produce effective reducts. The main idea here is to classify objects with minimal error. In this method, the relative classification error is calculated between the equivalence classes of condition and decision attributes.

The dynamic reduct-based method [3] is another rough set-based attribute reduction algorithm, which is based on the idea that the reducts obtained from an information system are sensitive to changes in the system. This method generates a large number of reducts by randomly removing objects from the original data. The reducts whose proportion of occurrence is more than a defined threshold are considered as the dynamic reducts. The main drawback of this method is that a predefined threshold value is required. Also, the generation of all reducts is computationally very costly. Recently, a distance measure-based approach is reported in [41] to explore the rough set boundary region for feature selection. However, all these approaches are computationally very costly. Different heuristic approaches based on rough set theory are also developed for feature selection [38, 63]. Combining rough sets and genetic algorithms, different algorithms have been proposed in [4, 52, 60] to discover optimal or close to optimal subset of features. However, most of the rough set-based feature selection methods proposed in [3, 4, 8, 50, 52, 60, 65] select the relevant or predictive features of a data set without considering the redundancy among them.

This chapter presents the rough set-based maximum relevance-maximum significance (RSMRMS) method, proposed by Maji and Paul in [37], to select a set of molecular descriptors for predicting biological activity of molecules. It employs rough sets to provide a means by which discrete valued data can be effectively reduced without the need for user-specified information. The RSMRMS method selects a subset of molecular descriptors from the whole feature set by maximizing both relevance and significance of the selected descriptors. The relevance and significance of the descriptors are calculated based on rough set theory. Hence, the only information required in the new feature selection method is in the form of equivalence partitions for each attribute, which can be automatically derived from the given data set. This avoids the need for domain experts to provide information on the data involved and ties in with the advantage of rough sets is that it requires no information other than the data set itself. The performance of the RSMRMS approach is compared with that of existing approaches using the R^2 statistic of support vector regression method. An important finding is that the RSMRMS approach is shown to be effective for selecting relevant and significant molecular descriptors from the QSAR data sets. The effectiveness of the RSMRMS method, along with a comparison with other related methods, is demonstrated on three QSAR data sets.

The structure of the rest of this chapter is as follows: Sect. 4.2 introduces the necessary notions of rough sets. The new feature selection method is described in Sect. 4.3 for predicting biological activity of molecules. A few case studies and a comparison with other related methods are presented in Sect. 4.4. Concluding remarks are given in Sect. 4.5.

4.2 Basics of Rough Sets

The theory of rough sets begins with the notion of an approximation space, which is a pair $< \mathbb{U}, \mathbb{A} >$, where $\mathbb{U} = \{x_1, \ldots, x_i, \ldots, x_n\}$ be a nonempty set, the universe of discourse, and $\mathbb{A}$ is a family of attributes, also called knowledge in the universe.

V is the value domain of $\mathbb{A}$ and f is an information function $f : \mathbb{U} \times \mathbb{A} \rightarrow V$. An approximation space is also called an information system [42].

Any subset $\mathbb{P}$ of knowledge $\mathbb{A}$ defines an equivalence, also called indiscernibility, relation $IND(\mathbb{P})$ on $\mathbb{U}$

$$IND(\mathbb{P}) = \{(x_i, x_j) \in \mathbb{U} \times \mathbb{U} | \forall a \in \mathbb{P}, f(x_i, a) = f(x_j, a)\}. \tag{4.1}$$

If $(x_i, x_j) \in IND(\mathbb{P})$, then x_i and x_j are indiscernible by attributes from $\mathbb{P}$. The partition of $\mathbb{U}$ generated by $IND(\mathbb{P})$ is denoted as

$$\mathbb{U}/IND(\mathbb{P}) = \{[x_i]_{\mathbb{P}} : x_i \in \mathbb{U}\} \tag{4.2}$$

where $[x_i]_{\mathbb{P}}$ is the equivalence class containing x_i. The elements in $[x_i]_{\mathbb{P}}$ are indiscernible or equivalent with respect to knowledge $\mathbb{P}$. Equivalence classes, also termed as information granules, are used to characterize arbitrary subsets of $\mathbb{U}$. The equivalence classes of $IND(\mathbb{P})$ and the empty set $\emptyset$ are the elementary sets in the approximation space $< \mathbb{U}, \mathbb{A} >$.

Given an arbitrary set $X \subseteq \mathbb{U}$, in general, it may not be possible to describe X precisely in $< \mathbb{U}, \mathbb{A} >$. One may characterize X by a pair of lower and upper approximations defined as follows [42]:

$$\underline{\mathbb{P}}(X) = \bigcup \{[x_i]_{\mathbb{P}} \mid [x_i]_{\mathbb{P}} \subseteq X\}; \tag{4.3}$$

and

$$\overline{\mathbb{P}}(X) = \bigcup \{[x_i]_{\mathbb{P}} \mid [x_i]_{\mathbb{P}} \cap X \neq \emptyset\}. \tag{4.4}$$

Hence, the lower approximation $\underline{\mathbb{P}}(X)$ is the union of all the elementary sets which are subsets of X, and the upper approximation $\overline{\mathbb{P}}(X)$ is the union of all the elementary sets which have a nonempty intersection with X. The tuple $< \underline{\mathbb{P}}(X), \overline{\mathbb{P}}(X) >$ is the representation of an ordinary set X in the approximation space $< \mathbb{U}, \mathbb{A} >$ or simply called the rough set of X. The lower (respectively, upper) approximation $\underline{\mathbb{P}}(X)$ (respectively, $\overline{\mathbb{P}}(X)$) is interpreted as the collection of those elements of $\mathbb{U}$ that definitely (respectively, possibly) belong to X. The lower approximation is also called positive region sometimes, denoted as $POS_{\mathbb{P}}(X)$. A set X is said to be definable or exact in $< \mathbb{U}, \mathbb{A} >$ iff $\underline{\mathbb{P}}(X) = \overline{\mathbb{P}}(X)$. Otherwise X is indefinable and termed as a rough set. $BN_{\mathbb{P}}(X) = \overline{\mathbb{P}}(X) \setminus \underline{\mathbb{P}}(X)$ is called a boundary set.

Definition 4.1 An information system $< \mathbb{U}, \mathbb{A} >$ is called a decision table if the attribute set $\mathbb{A} = \mathbb{C} \cup \mathbb{D}$, where $\mathbb{C}$ and $\mathbb{D}$ represent the condition and decision attribute sets, respectively. The dependency between $\mathbb{C}$ and $\mathbb{D}$ can be defined as

$$\gamma_{\mathbb{C}}(\mathbb{D}) = \frac{|POS_{\mathbb{C}}(\mathbb{D})|}{|\mathbb{U}|} \tag{4.5}$$

where $POS_{\mathbb{C}}(\mathbb{D}) = \cup \underline{\mathbb{C}} X_i$, X_i is the ith equivalence class induced by $\mathbb{D}$ and $|\cdot|$ denotes the cardinality of a set.

Let $\mathbb{I} =< \mathbb{U}, \mathbb{A} >$ be a decision table, where $\mathbb{U} = \{x_1, \ldots, x_7\}$ is a nonempty set of finite objects, the universe, and $\mathbb{A} = \mathbb{C} \cup \mathbb{D}$ is a nonempty finite set of attributes. Here, $\mathbb{C} = \{\mathscr{A}_1, \mathscr{A}_2\}$ and $\mathbb{D} = \{\text{Walk}\}$ are the sets of condition and decision attributes, respectively.

$x_i \in \mathbb{U}$	$\mathscr{A}_1$	$\mathscr{A}_2$	Walk
x_1	$16-30$	50	yes
x_2	$16-30$	0	no
x_3	$31-45$	$1-25$	no
x_4	$31-45$	$1-25$	yes
x_5	$46-60$	$26-49$	no
x_6	$16-30$	$26-49$	yes
x_7	$46-60$	$26-49$	no

$IND(\{\mathscr{A}_1\})$ creates the following partition of $\mathbb{U}$:

$$\mathbb{U}/IND(\{\mathscr{A}_1\}) = \{\{x_1, x_2, x_6\}, \{x_3, x_4\}, \{x_5, x_7\}\}$$

as the objects x_1, x_2, and x_6 are indiscernible with respect to the condition attribute set $\{\mathscr{A}_1\}$. Similarly, the partition of $\mathbb{U}$ generated by the condition attribute set $\{\mathscr{A}_2\}$ is given by:

$$\mathbb{U}/IND(\{\mathscr{A}_2\}) = \{\{x_1\}, \{x_2\}, \{x_3, x_4\}, \{x_5, x_6, x_7\}\}$$

and the partition of $\mathbb{U}$ generated by the condition attribute set $\{\mathscr{A}_1, \mathscr{A}_2\}$ is as follows:

$$\mathbb{U}/IND(\{\mathscr{A}_1, \mathscr{A}_2\}) = \{\{x_1\}, \{x_2\}, \{x_3, x_4\}, \{x_5, x_7\}, \{x_6\}\}.$$

Similarly, the partition of $\mathbb{U}$ generated by the decision attribute set $\{\text{Walk}\}$ is given by:

$$\mathbb{U}/IND(\mathbb{D}) = \mathbb{U}/IND(\{\text{Walk}\}) = \{\{x_1, x_4, x_6\}, \{x_2, x_3, x_5, x_7\}\}.$$

The positive region contains all objects of $\mathbb{U}$ that can be classified to classes of $\mathbb{U}/IND(\mathbb{D})$ using the knowledge in attributes $\mathbb{C}$. Hence, for the above example, the positive region is as follows:

$$POS_{\mathbb{C}}(\mathbb{D}) = \bigcup \{\phi, \{x_1\}, \{x_2\}, \{x_5, x_7\}, \{x_6\}\} = \{x_1, x_2, x_5, x_6, x_7\}.$$

The dependency between the attributes $\mathbb{C}$ and $\mathbb{D}$ is, therefore, given by

$$\gamma_{\mathbb{C}}(\mathbb{D}) = \frac{5}{7}.$$

An important issue in data analysis is discovering dependency between attributes. Intuitively, a set of attributes $\mathbb{D}$ depends totally on a set of attributes $\mathbb{C}$, denoted as $\mathbb{C} \Rightarrow \mathbb{D}$, if all attribute values from $\mathbb{D}$ are uniquely determined by values of attributes from $\mathbb{C}$. If there exists a functional dependency between values of $\mathbb{D}$ and $\mathbb{C}$, then $\mathbb{D}$ depends totally on $\mathbb{C}$. Dependency can be defined in the following way:

Definition 4.2 For $\mathbb{C}, \mathbb{D} \subseteq \mathbb{A}$, it is said that $\mathbb{D}$ depends on $\mathbb{C}$ in a degree κ $(0 \leq \kappa \leq 1)$, denoted as $\mathbb{C} \Rightarrow_\kappa \mathbb{D}$, if

$$\kappa = \gamma_{\mathbb{C}}(\mathbb{D}) = \frac{|POS_{\mathbb{C}}(\mathbb{D})|}{|\mathbb{U}|}. \tag{4.6}$$

If $\kappa = 1$, $\mathbb{D}$ depends totally on $\mathbb{C}$, if $0 < \kappa < 1$, $\mathbb{D}$ depends partially (in a degree κ) on $\mathbb{C}$, and if $\kappa = 0$, then $\mathbb{D}$ does not depend on $\mathbb{C}$.

To what extent an attribute is contributing to calculate the dependency on decision attribute can be calculated by the significance of that attribute. The change in dependency when an attribute is removed from the set of condition attributes, is a measure of the significance of the attribute. The higher the change in dependency, the more significant the attribute is. If the significance is 0, then the attribute is dispensable.

Definition 4.3 Given $\mathbb{C}$, $\mathbb{D}$ and an attribute $\mathscr{A} \in \mathbb{C}$, the significance of the attribute $\mathscr{A}$ is defined as follows:

$$\sigma_{\mathbb{C}}(\mathbb{D}, \mathscr{A}) = \gamma_{\mathbb{C}}(\mathbb{D}) - \gamma_{\mathbb{C}-\mathscr{A}}(\mathbb{D}). \tag{4.7}$$

Considering the above example, let $\mathbb{C} = \{\mathscr{A}_1, \mathscr{A}_2\}$ and $\mathbb{D} = \{\text{Walk}\}$. The significance values of two attributes $\mathscr{A}_1$ and $\mathscr{A}_2$ are as follows:

$$\sigma_{\mathbb{C}}(\mathbb{D}, \mathscr{A}_1) = \gamma_{\mathbb{C}}(\mathbb{D}) - \gamma_{\mathbb{C}-\mathscr{A}_1}(\mathbb{D}) = \frac{5}{7} - \frac{2}{7} = \frac{3}{7},$$
$$\sigma_{\mathbb{C}}(\mathbb{D}, \mathscr{A}_2) = \gamma_{\mathbb{C}}(\mathbb{D}) - \gamma_{\mathbb{C}-\mathscr{A}_2}(\mathbb{D}) = \frac{5}{7} - \frac{2}{7} = \frac{3}{7}.$$

4.3 Rough Set-Based Molecular Descriptor Selection Algorithm

The main objective of the current research is to build a method that can effectively find out biological activity values of molecules provided with their molecular descriptors. In effect, it can help to decide which features of a molecule give rise to its overall activity and help to make modified compounds with enhanced properties.

In general, the QSAR data set may contain a number of insignificant molecular descriptors. The presence of such irrelevant and insignificant molecular descriptors can produce inappropriate information. A standard descriptor set is the one that has high relevance with the activity values and high significance in the feature set. The molecular descriptors with high relevance are expected to predict the biological

activity effectively. However, if insignificant descriptors are present in the subset, they may reduce the prediction capability. A feature set with high relevance and high significance enhances the predictive capability. Accordingly, a measure is required that can enhance the effectiveness of descriptors. In this chapter, the theory of rough sets is used to select the relevant and significant molecular descriptors from the QSAR data set based on maximum relevance-maximum significance (MRMS) criterion.

4.3.1 Maximum Relevance-Maximum Significance Criterion

Let $\mathbb{U} = \{x_1, \ldots, x_i, \ldots, x_n\}$ be the set of n molecules and $\mathbb{M} = \{\mathbb{M}_1, \ldots, \mathbb{M}_j, \ldots, \mathbb{M}_m\}$ is the set of m molecular descriptors of a QSAR data set. These molecules and descriptors form a table $\mathscr{T} = \{w_{ij} | i = 1, \ldots, n, j = 1, \ldots, m\}$, where $w_{ij} \in \Re$ is the measured value of the molecular descriptor $\mathbb{M}_j$ in the molecule x_i. Let $\mathbb{B} = \{\mathscr{B}_1, \ldots, \mathscr{B}_i, \ldots, \mathscr{B}_n\}$ be the set of biological activity values of n molecules, where $\mathscr{B}_i \in \Re$ is the activity value of the molecule x_i. Hence, in terms of rough set theory, a QSAR data set can be considered as a decision table $\mathbb{I} =< \mathbb{U}, \mathbb{M} \cup \mathbb{B} >$, where $\mathbb{M}$ and $\mathbb{B}$ play the role of condition and decision attribute sets, respectively. However, the continuous values are discretized to compute the relevance and significance of descriptors using rough sets.

Let $\mathbb{S}$ be the set of selected descriptors with cardinality $d < m$. Define $\hat{f}(\mathbb{M}_i, \mathbb{B})$ as the relevance of the descriptor $\mathbb{M}_i$ with respect to the response variable or biological activity value $\mathbb{B}$ while $\tilde{f}(\mathbb{M}_i, \mathbb{M}_j)$ as the significance of the descriptor $\mathbb{M}_j$ with respect to the already selected descriptor $\mathbb{M}_i$. The total relevance of all selected descriptors is, therefore, given by

$$\mathscr{J}_{\text{relev}} = \sum_{\mathbb{M}_i \in \mathbb{S}} \hat{f}(\mathbb{M}_i, \mathbb{B}). \tag{4.8}$$

The task of descriptor or feature selection is to find a descriptor subset $\mathbb{S} \subseteq \mathbb{M}$ that maximizes the objective function $\mathscr{J}_{\text{relev}}$. In terms of rough set theory, the relevance $\hat{f}(\mathbb{M}_i, \mathbb{B})$ of a molecular descriptor $\mathbb{M}_i$ with respect to the biological activity $\mathbb{B}$ can be calculated using (4.6), that is,

$$\mathscr{J}_{\text{relev}} = \sum_{\mathbb{M}_i \in \mathbb{S}} \gamma_{\mathbb{M}_i}(\mathbb{B}). \tag{4.9}$$

However, it is likely that descriptors selected according to the above criterion could have rich redundancy, that is, the dependency among these descriptors could be large. When two molecular descriptors highly depend on each other, the respective biological activity prediction power would not change much if one of them were removed. It follows that one descriptor is dispensable with respect to other. The significance criterion defined in (4.7) is able to find out the dispensable descriptors.

If the significance of a descriptor with respect to another descriptor is 0, then the descriptor is dispensable [42]. Therefore, the significance criterion can be added to select mutually exclusive descriptors. The total significance among the selected descriptors is

$$\mathscr{J}_{\text{signf}} = \sum_{\mathbb{M}_i \neq \mathbb{M}_j \in \mathbb{S}} \tilde{f}(\mathbb{M}_i, \mathbb{M}_j). \tag{4.10}$$

In the RSMRMS method, the significance $\tilde{f}(\mathbb{M}_i, \mathbb{M}_j)$ of the descriptor $\mathbb{M}_j$ with respect to the already selected descriptor $\mathbb{M}_i$ is computed using (4.7). That is,

$$\mathscr{J}_{\text{signf}} = \sum_{\mathbb{M}_i \neq \mathbb{M}_j \in \mathbb{S}} \sigma_{\mathbb{M}_i \cup \mathbb{M}_j}(\mathbb{B}, \mathbb{M}_j). \tag{4.11}$$

Therefore, the problem of selecting a set $\mathbb{S}$ of d relevant and significant descriptors from the whole set $\mathbb{M}$ of m descriptors is equivalent to maximize both $\mathscr{J}_{\text{relev}}$ and $\mathscr{J}_{\text{signf}}$, that is, to maximize the objective function $\mathscr{J}$, where

$$\mathscr{J} = \mathscr{J}_{\text{relev}} + \mathscr{J}_{\text{signf}}, \tag{4.12}$$

that is,

$$\mathscr{J} = \sum_{\mathbb{M}_i \in \mathbb{S}} \gamma_{\mathbb{M}_i}(\mathbb{B}) + \sum_{\mathbb{M}_i \neq \mathbb{M}_j \in \mathbb{S}} \sigma_{\mathbb{M}_i \cup \mathbb{M}_j}(\mathbb{B}, \mathbb{M}_j). \tag{4.13}$$

Obviously, when d equals 1, the solution is the molecular descriptor that maximizes $\hat{f}(\mathbb{M}_i, \mathbb{B})$; $(1 \leq i \leq m)$. When $d > 1$, a simple incremental search scheme is to add one descriptor at one time. This type of selection is called the first-order incremental search. By definition of first-order search, it is assumed that the set of $(d-1)$ descriptors has already been obtained. The task is to select the optimal dth descriptor $\mathbb{M}_j$ from the remaining descriptors of the set $\mathbb{M}$ that contributes to the largest increase of the following condition:

$$\hat{f}(\mathbb{M}_j, \mathbb{B}) + \frac{1}{|\mathbb{S}|} \sum_{\mathbb{M}_i \in \mathbb{S}} \tilde{f}(\mathbb{M}_i, \mathbb{M}_j), \qquad \text{where } |\mathbb{S}| = d - 1. \tag{4.14}$$

Hence, the following greedy algorithm is used to select relevant and significant descriptors from a QSAR data set.

1. Initialize $\mathbb{M} \leftarrow \{\mathbb{M}_1, \ldots, \mathbb{M}_i, \ldots, \mathbb{M}_m\}, \mathbb{S} \leftarrow \emptyset$.
2. Calculate the relevance value $\hat{f}(\mathbb{M}_i, \mathbb{B})$ of each descriptor $\mathbb{M}_i \in \mathbb{M}$ with respect to the biological activity $\mathbb{B}$.
3. Select the descriptor $\mathbb{M}_i$ as the most relevant descriptor that has highest relevance $\hat{f}(\mathbb{M}_i, \mathbb{B})$. In effect, $\mathbb{M}_i \in \mathbb{S}$ and $\mathbb{M} = \mathbb{M} \setminus \mathbb{M}_i$.
4. Repeat the following two steps until the desired number of descriptors is selected.

5. Calculate the significance of each of the remaining descriptors of $\mathbb{M}$ with respect to the already selected descriptors of $\mathbb{S}$.
6. From the remaining descriptors of $\mathbb{M}$, select descriptor $\mathbb{M}_j$ that maximizes

$$\hat{f}(\mathbb{M}_j, \mathbb{B}) + \frac{1}{|\mathbb{S}|} \sum_{\mathbb{M}_i \in \mathbb{S}} \tilde{f}(\mathbb{M}_i, \mathbb{M}_j). \tag{4.15}$$

   As a result of that, $\mathbb{M}_j \in \mathbb{S}$ and $\mathbb{M} = \mathbb{M} \setminus \mathbb{M}_j$.
7. Stop.

In the RSMRMS method, the relevance $\hat{f}(\mathbb{M}_i, \mathbb{B})$ of a molecular descriptor $\mathbb{M}_i$ with respect to the biological activity $\mathbb{B}$ is calculated using (4.6), while the significance $\tilde{f}(\mathbb{M}_i, \mathbb{M}_j)$ of the descriptor $\mathbb{M}_j$ with respect to the already selected descriptor $\mathbb{M}_i$ is computed using (4.7).

4.3.2 Computational Complexity

The rough set theory-based feature selection method (RSMRMS) has low computational complexity with respect to the number of descriptors in the original data set. The computation of the relevance of m descriptors is carried out in step 2 of the RSMRMS algorithm, which has $\mathcal{O}(m)$ time complexity. The selection of most relevant descriptor from the set of m descriptors, that is step 3, has also a complexity $\mathcal{O}(m)$. There is only one loop in the RSMRMS method, which is executed $(d-1)$ times, where d represents the number of selected features. Each iteration of the loop takes only a constant amount of time. The complexity to calculate the significance of a descriptor with respect to the already selected descriptors is $\mathcal{O}(\acute{m})$, where $\acute{m}$ is the cardinality of the already selected descriptor set. In effect, the selection of a set of d relevant and significant descriptors from the whole set of m descriptors using the new first-order incremental search method has an overall computational complexity of $(\mathcal{O}(m) + \mathcal{O}(d\acute{m})) = \mathcal{O}(m)$ as $d, \acute{m} << m$.

4.3.3 Generation of Equivalence Classes

In QSAR data set, the molecular descriptor values as well as the biological activity values of different molecules are continuous. Hence, to measure both relevance and significance of molecular descriptors using rough set theory, the continuous descriptor values of a molecule are usually divided into several discrete partitions to generate equivalence classes. The discretization method reported in [34] is employed to discretize the continuous descriptor values. The values of a descriptor or an attribute are discretized using mean μ and standard deviation σ computed over n values of that attribute: any value larger than $(\mu + \frac{\sigma}{2})$ is transformed to state 1; any value between

$(\mu - \frac{\sigma}{2})$ and $(\mu + \frac{\sigma}{2})$ is transformed to state 0; any value smaller than $(\mu - \frac{\sigma}{2})$ is transformed to state -1 [34]. The equivalence classes are then generated to compute both relevance and significance of molecular descriptors.

4.4 Experimental Results

The performance of the rough set-based maximum relevance-maximum significance (RSMRMS) [37] method is extensively studied and compared with that of some existing algorithms. The source code of the RSMRMS algorithm, written in C language, is available at http://www.isical.ac.in/~bibl/results/rsmrms/rsmrms.html. All other algorithms are also implemented in C language and run in LINUX environment having machine configuration Pentium IV, 2.8 GHz, 1 MB cache, and 512 MB RAM. To analyze the performance of different algorithms, the experimentation is done on three QSAR data sets. The major metric for evaluating the performance of different algorithms is the R^2 statistic of support vector regression method.

4.4.1 Description of QSAR Data Sets

In this chapter, following three QSAR data sets are used that are available at http://www.cheminformatics.org.

4.4.1.1 Steroid Data Set

This data set contains 31 steroid molecules presented in MOL format, which is used in cheminformatics applications for storing atomic coordinates, chemical bond information, and metadata of the 3D structure of a single chemical compound in plain text tabular format. The $\log(1/k)$ values of these molecules are also given. All these molecules are categorized into three activity classes. Among them, 11 are reported as high activity molecules, 9 moderate and rest 11 as lowest activity molecules.

4.4.1.2 Small Dopamine Data Set

It contains 26 dopamine molecules given in MOL format. The biological activity of these molecules is also available.

4.4.1.3 Large Dopamine Data Set

This data set consists of 116 dopamine molecules that are given along with their molecular descriptors in binary form. The continuous valued biological activity of each molecule is also given.

Both steroid and small dopamine data sets are available in MOL format. The molecular descriptors of these data sets are obtained using MODEL software [31], which calculates approximately 4000 molecular descriptors for each molecule. The calculated descriptors cover different aspects of the molecular structure including topological, electronic, constitutional, geometrical, and physical descriptors.

4.4.2 Support Vector Regression Method

The support vector machine (SVM) [59] is a relatively new and promising classification and regression method. It is a margin classifier that draws an optimal hyperplane in the feature vector space; this defines a boundary that maximizes the margin between data samples in different classes, therefore leading to good generalization properties. A key factor in the SVM is to use kernels to construct nonlinear decision boundary. In this work, radial basis function kernels are used. The source code of the SVM is downloaded from http://www.csie.ntu.edu.tw/~cjlin/libsvm. A brief introduction of the SVM is also reported in Chap. 3.

The performance of the SVM is analyzed using R^2 statistic or coefficient of determination value. The R^2 statistic tells about the goodness of fit of a model and how well a regression approximates its attributes. The value of R^2 statistic ranges from 0 to 1. The near the value reaches to 1, the better is the approximation. The R^2 statistic can be calculated as follows:

$$R^2 = 1 - \frac{SS_{\text{err}}}{SS_{\text{tot}}}; \tag{4.16}$$

where

$$SS_{\text{tot}} = \Sigma_i (y_i - \bar{y})^2 \tag{4.17}$$

and

$$SS_{\text{err}} = \Sigma_i (y_i - f_i)^2 \tag{4.18}$$

represent the total sum of squares, which is proportional to the sample variance, and the sum of squared errors, also called the residual sum of squares, respectively. Here, $\bar{y}$ represents the mean of the observed data, while y_i and f_i are the ith observed and modeled or predicted values, respectively.

Table 4.1 Performance for different number of equivalence classes

Data Set	Experiment	$c = 2$	$c = 3$	$c = 5$
Steroid	10-fold CV	0.12	0.89	0.84
	LOOCV	0.33	0.88	0.84
Small	10-fold CV	0.18	0.37	0.24
Dopamine	LOOCV	0.39	0.45	0.27
Large	10-fold CV	0.52	0.52	0.52
Dopamine	LOOCV	0.53	0.53	0.53

4.4.3 Optimum Number of Equivalence Classes

In QSAR data set, both molecular descriptors and biological activity values are continuous. Hence, to measure the relevance and significance of descriptors using rough set theory, the continuous values have to be divided into several discrete partitions to generate equivalence classes. In the RSMRMS method, the continuous values are discretized into three ($c = 3$) states as per the procedure reported in Sect. 4.3.3.

In order to establish the effectiveness of three ($c = 3$) state discretization procedure, the extensive experiments are carried out on different QSAR data sets. The performance of the RSMRMS method for $c = 3$ is compared with that for $c = 2$ and 5. For $c = 2$, any value larger than mean is transformed to one state, while others to another state. On the other hand, for $c = 5$, the intermediate state of $c = 3$ is partitioned into three states, while other two states remain unaltered, therefore leading to total five states. Table 4.1 reports the comparative performance of the RSMRMS method for $c = 2$, 3, and 5 with respect to the R^2 statistic of the SVM. To compute the R^2 statistic, both leave-one-out cross-validation (LOOCV) and 10-fold cross-validation (CV) are performed on each QSAR data set. All the results reported in Table 4.1 establish the fact that the performance of the rough set-based feature selection method, that is, RSMRMS method, is significantly better in case of $c = 3$ than that of $c = 2$ and 5.

4.4.4 Performance Analysis

The experimental results on three QSAR data sets are presented in Figs. 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8 and 4.9. Subsequent discussions analyze the results with respect to the R^2 statistic of the SVM. To compute the R^2 statistic of the SVM, both LOOCV and 10-fold CV are performed on each QSAR data set. The number of molecular descriptors selected ranges from 1 to 50.

Figure 4.1 presents the performance of the RSMRMS method on steroid molecules obtained by both 10-fold CV and LOOCV, while Figs. 4.2 and 4.3 depict that for small and large dopamine molecules, respectively. From Fig. 4.1, it is seen that as the number of selected descriptors of steroid molecules ranges from 1 to 15, the R^2 statistic of the SVM fluctuates in case of both 10-fold CV and LOOCV. It indicates

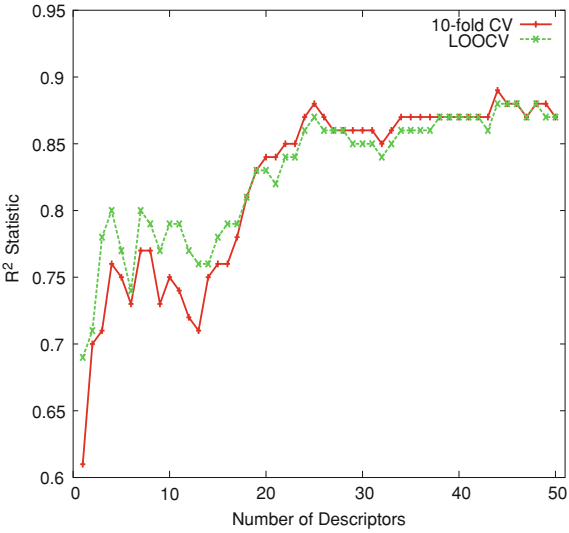

Fig. 4.1 Results on steroid molecules obtained by 10-fold CV and LOOCV

that the RSMRMS method gets stuck into local minima of the search space for this range. However, the R^2 statistic continuously increases with the increase in number of selected descriptors for more than 15. Finally, the RSMRMS method attains its maximum R^2 statistic of 0.88 and 0.89 using only 44 descriptors for the LOOCV and 10-fold CV, respectively. That is, the RSMRMS method is able to find out an optimum or near to optimum solution using 44 descriptors for both 10-fold CV and LOOCV. On the other hand, from Fig. 4.2, it can be seen that in case of small dopamine molecules, two most relevant and significant descriptors are sufficient to achieve the maximum R^2 statistic values of 0.45 and 0.37 of the RSMRMS method for the LOOCV and 10-fold CV, respectively. Finally, Fig. 4.3 depicts the results for large dopamine molecules. From the results presented in Fig. 4.3, it is seen that the RSMRMS method attains maximum R^2 statistic of 0.53 with 9 descriptors using the LOOCV, while for 10-fold CV, the best R^2 statistic is 0.52 with same number of descriptors. In other words, the RSMRMS method is able to find out optimum or near to optimum solutions using 2 and 9 molecular descriptors for small and large dopamine molecules, respectively.

Figures 4.4, 4.5 and 4.6 present the comparative performance analysis of the RSMRMS method and one of the most popular rough set-based algorithms, called quick reduct algorithm [8]. All the results are reported for three QSAR data sets based on the LOOCV. The actual and obtained biological activity values of different molecules for three QSAR data sets are reported for comparison. The R^2 statistic values of quick reduct algorithm are 0.82, 0.45, and 0.56 for steroid, small dopamine, and large dopamine molecules, respectively. For 10-fold CV, the R^2 statistic values of quick reduct algorithm are 0.83, 0.37, and 0.52 on steroid, small dopamine, and large dopamine, respectively. From the results reported in Figs. 4.4, 4.5 and 4.6, it is seen that the performance of the RSMRMS method is better than quick reduct algorithm

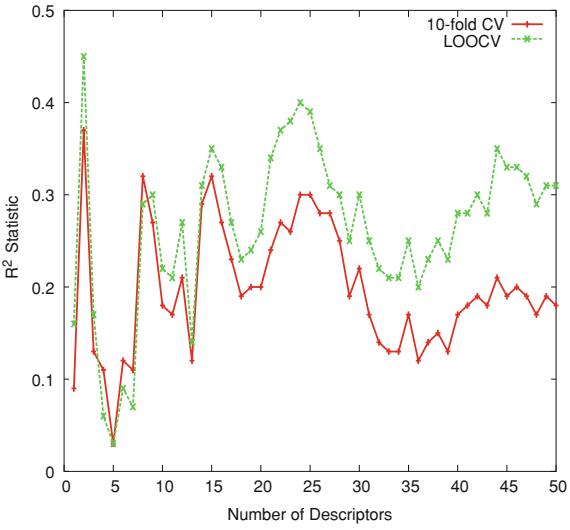

Fig. 4.2 Results on small dopamine obtained by 10-fold CV and LOOCV

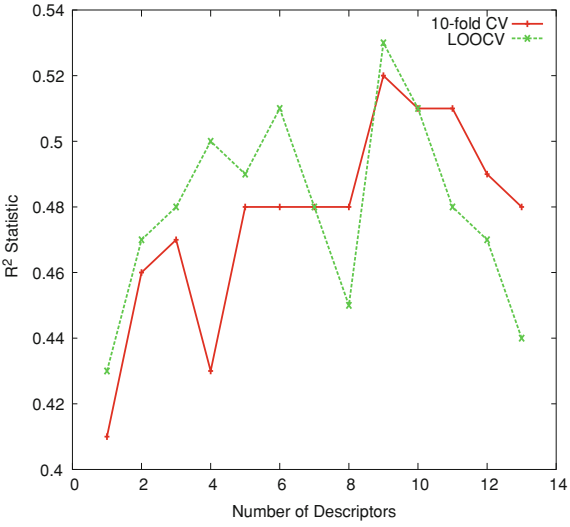

Fig. 4.3 Results on large dopamine obtained by 10-fold CV and LOOCV

in case of steroid data set and comparable with quick reduct algorithm for both small and large dopamine molecules. In this regard, it should be noted that another rough set-based algorithm, called discernibility matrix-based method [50], attains the R^2 statistic values of 0.79, 0.43, and 0.39 for steroid, small dopamine, and large dopamine molecules, respectively, using 10-fold CV, while the corresponding values for the LOOCV are 0.79, 0.61, and 0.41, respectively. However, as the computational complexity of both quick reduct method [8] and discernibility matrix-based method

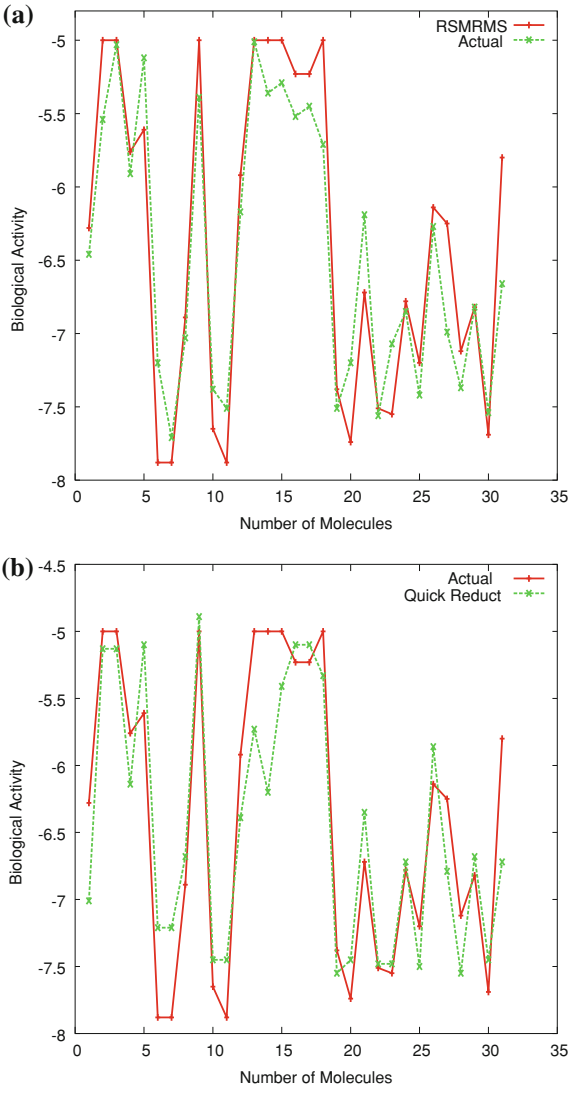

Fig. 4.4 Results for steroid molecules obtained by leave-one-out cross-validation. **a** RSMRMS method. **b** Quick reduct algorithm

[50] is very high, they require significantly higher execution time compared to that of the RSMRMS algorithm.

Table 4.2 compares the execution time, in milli second, of the RSMRMS algorithm and that of quick reduct algorithm [8] and discernibility matrix-based method [50] for three QSAR data sets. From the results reported in Table 4.2, it is seen that the execution time required for the RSMRMS algorithm is significantly lower than that of other two algorithms, irrespective of the data sets used. As the computational complexity of both quick reduct algorithm and discernibility matrix-based

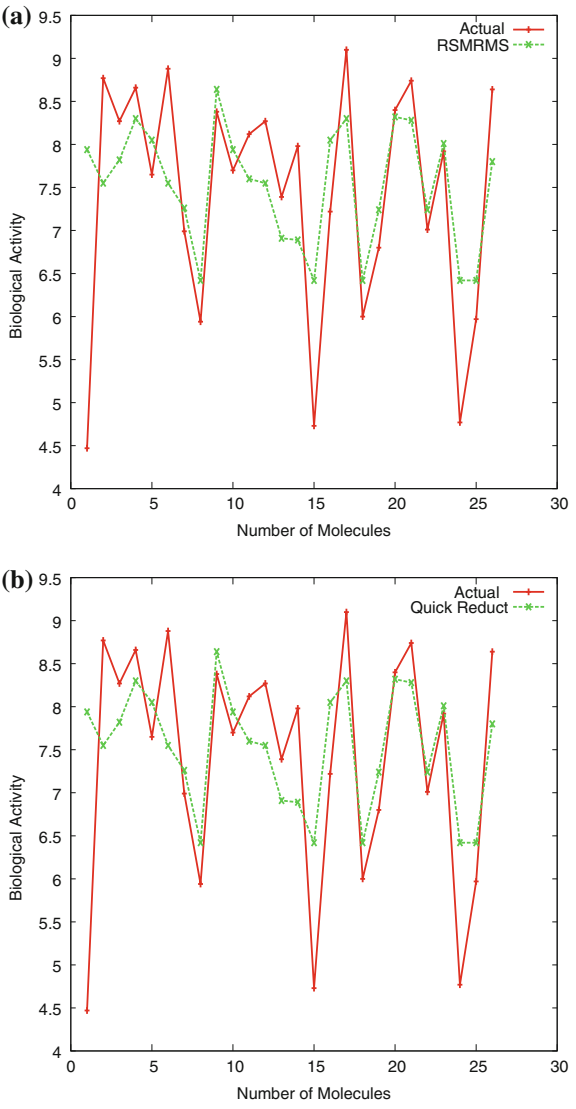

Fig. 4.5 Results for small dopamine molecules obtained by leave-one-out cross-validation. **a** RSMRMS method. **b** Quick reduct algorithm

method is exponential in nature [8, 50], they require significantly higher execution time compared to that of the RSMRMS algorithm. The significantly lesser execution time of the RSMRMS algorithm is achieved due to its low computational complexity.

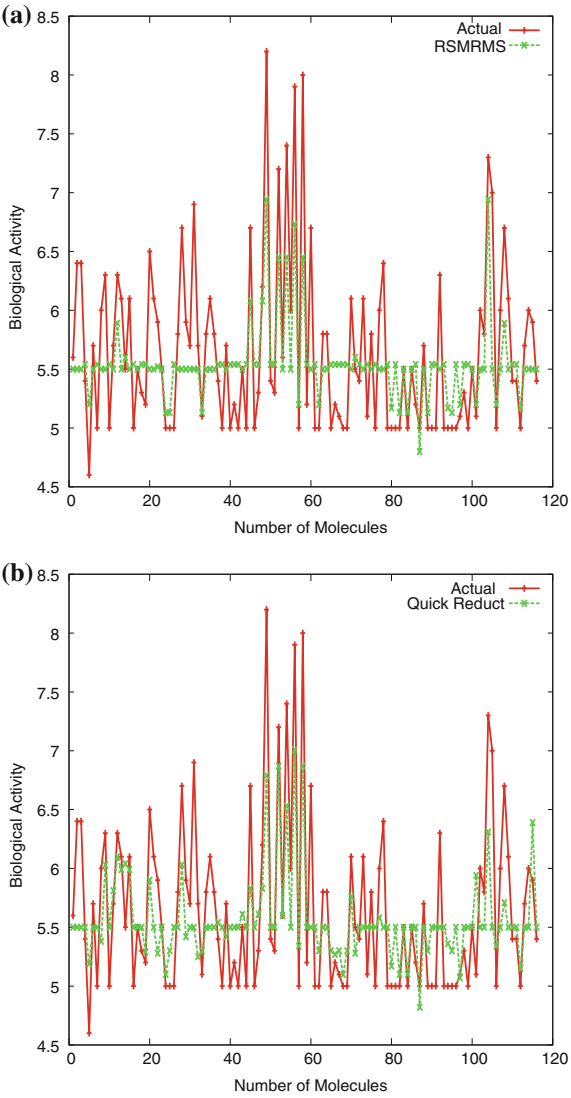

Fig. 4.6 Results for large dopamine molecules obtained by leave-one-out cross-validation. **a** RSMRMS method. **b** Quick reduct algorithm

4.4.5 Comparative Performance Analysis

The RSMRMS method performs significantly better than different existing QSAR methods. To establish the superiority of the RSMRMS method, extensive experimentation is carried out on different QSAR data sets. Figure 4.7 presents the predicted biological activity values of the RSMRMS method and Compass [19], an well-known existing QSAR model, along with the actual activity values. Results

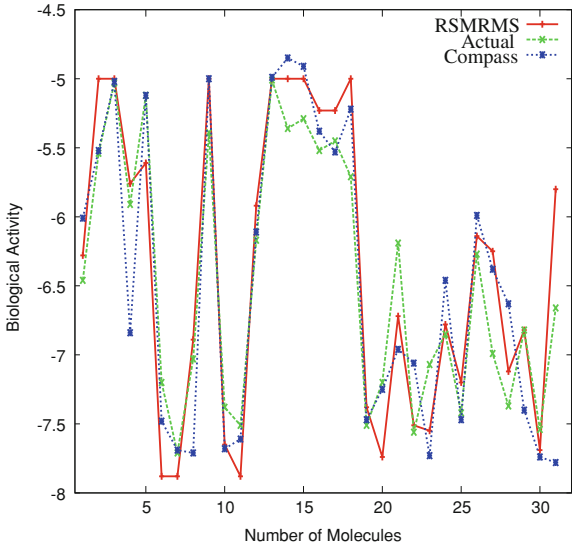

Fig. 4.7 Results of RSMRMS and compass on steroid molecules using LOOCV

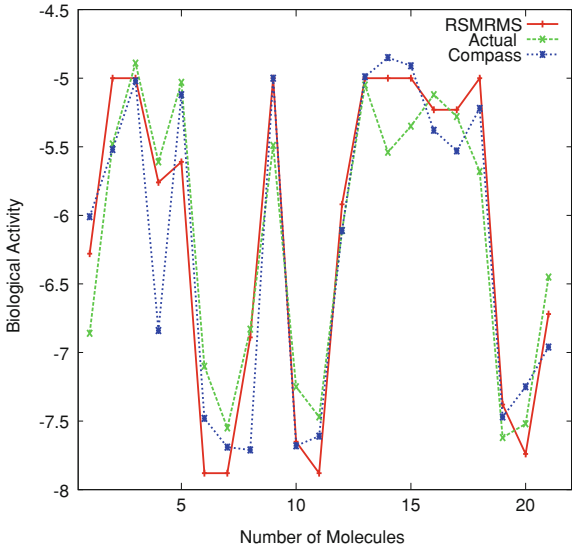

Fig. 4.8 Results of RSMRMS and compass on 21 training steroid molecules

are reported based on the LOOCV. The R^2 statistic values corresponding to the RSMRMS method and Compass are 0.89 and 0.79, respectively. Next, the steroid data set is divided into two sets: training set of 21 molecules and test set of 10 molecules. The LOOCV results of 21 molecules obtained by the RSMRMS method as well as two well-known existing approaches, namely, Compass [19] and CoMFA [57] are reported in Table 4.3. Figures 4.8 and 4.9 depict the actual and predicted values of the

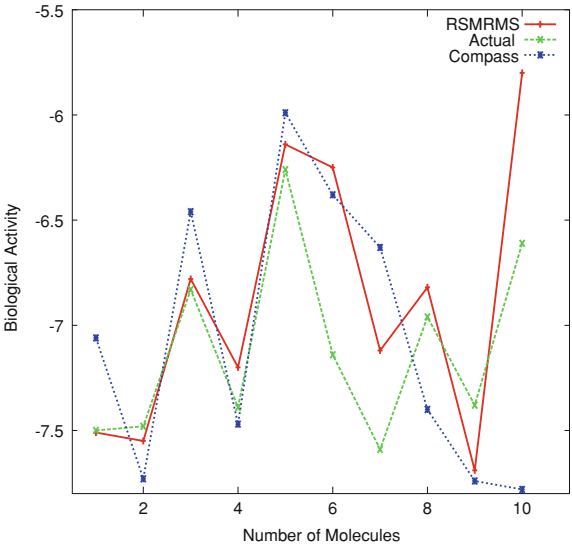

Fig. 4.9 Results of RSMRMS and compass on 10 test steroid molecules

Table 4.2 Execution time of different algorithms

Data Set	Quick reduct	Discernibility matrix	RSMRMS
Steroid	383253	55061	3498
Small dopamine	350015	54044	4299
Large dopamine	487735	35027	1755

Table 4.3 Result on training set of steroid data

	Methods	R^2 statistic
Existing	CoMFA	0.69
Models	Compass	0.89
RSMRMS	RSMRMS	0.97

RSMRMS method and Compass [19] for 21 training and 10 test steroid molecules, respectively. A detailed comparison of the RSMRMS method with other existing 3D QSAR methods, namely, Compass [19], MS-WHIM [5], PARM [6], TQSAR [47], SOMFA [48], EVA [57], CoMFA [57], COMSA [43], MEDV [33], QS-SM [1], and EEVA [56], is presented in Table 4.4 on test set of steroid data, that is, molecules 22 to 31.

From the R^2 statistic reported in Tables 4.3 and 4.4, along with the results reported in Figs. 4.7, 4.8 and 4.9, it can be seen that the RSMRMS method outperforms different existing QSAR approaches in case of steroid data set. Also, the RSMRMS method predicts biological activity of 21 training and 10 test molecules significantly better than the Compass [19]. Moreover, the model building phase of Compass takes about 1 minute per molecule for steroid data set [19], which is significantly higher than that of the RSMRMS method.

Table 4.4 Result on test set of steroid data

	Methods	R^2 statistic
Existing Models	Compass	0.16
	MS-WHIM	0.28
	PARM	0.33
	TQSAR	0.16
	SOMFA	0.20
	EVA	0.36
	CoMFA	0.25
	COMSA	0.09
	MEDV	0.45
	QS-SM	0.36
	EEVA	0.36
RSMRMS	RSMRMS	0.67

Among 2901 molecular descriptors of steroid data set, 44 relevant and significant descriptors obtained using the RSMRMS method can predict biological activity values of steroid molecules accurately. All these 44 descriptors can be grouped into one of the following four descriptor types, namely, topological, geometrical, electronic, and charge. By analyzing the R^2 statistic values of steroid data set, one can deduce that topological, geometrical, electronic, and charge descriptors are favorably affect the biological activities of these molecules, while thermodynamic and constitutional descriptors can make adverse affect on biological activities.

Finally, the 10-fold CV result of the RSMRMS method for large dopamine data is compared with the existing approach Boosting of Sventik et al. [53]. While the RSMRMS method achieves the R^2 value of 0.52 with 9 attributes, the best result obtained by the Boosting method is 0.48, that is, the RSMRMS method performs significantly better than the existing method.

4.5 Conclusion and Discussion

This chapter introduces a new feature selection algorithm based on rough set theory, called RSMRMS, in order to identify relevant and significant molecular descriptors from high dimensional QSAR data sets. It presents the results of selecting effective molecular descriptors for predicting biological activity of molecules. The maximum relevance-maximum significance (MRMS) criterion is used to select the molecular descriptors. The performance of the RSMRMS method is evaluated by the R^2 statistic of support vector regression method. For all data sets, significantly better results are found for the RSMRMS method compared to different existing QSAR models. The results obtained on real data sets demonstrate that the RSMRMS method can bring a remarkable improvement on descriptor selection problem, and therefore, it can be a promising alternative to existing QSAR models for prediction of biological activity of molecules. All the results reported in this chapter demonstrate the feasibility and effectiveness of the RSMRMS method. The RSMRMS method is capable of iden-

tifying effective molecular descriptors that may contribute to revealing underlying molecular structures, providing a useful tool for exploratory analysis of QSAR data sets.

However, there are usually real-valued data and fuzzy information in real-world applications. In rough set theory, the real-valued features are divided into several discrete partitions and the dependency or quality of approximation of a feature is calculated. The inherent error that exists in discretization process is of major concern in the computation of the dependency of real-valued features. A fuzzy set-based discretization method is reported in Chap. 7 to generate equivalence classes for continuous valued features required to compute the dependency using rough set theory. Combining fuzzy sets and rough sets provides an important direction in reasoning with uncertainty for real-valued data sets. Both fuzzy sets and rough sets provide a mathematical framework to capture uncertainties associated with the data [11, 12]. They are complementary in some aspects. The generalized theories of rough-fuzzy sets and fuzzy-rough sets have been applied successfully to feature selection of real valued data [20, 21, 23, 55, 61]. Both fuzzy-rough sets [16, 22] and neighborhood rough sets [17, 32] can handle continuous valued attributes or features without any discretization. Jensen and Shen [20, 21] introduced the fuzzy-rough quick reduct algorithm for feature selection or dimensionality reduction of real-valued data sets. In [18], Hu et al. have used the concept of fuzzy equivalence relation matrix to compute entropy and mutual information in fuzzy approximation spaces, which can be used for feature selection of real-valued data sets. A feature selection method is proposed in [36], which employs fuzzy-rough sets and f-information measures to provide a means by which discrete or real-valued noisy data, or a mixture of both, can be effectively reduced without the need for user-specified information. Recently, Maji and Garai introduced a feature selection method in [35], integrating judiciously the merits of MRMS criterion and fuzzy-rough sets, to select relevant and significant real-valued features from high dimensional noisy data set.

Through the current investigations and experiments, the potential utility of rough sets and MRMS criterion for molecular descriptor selection from QSAR data sets are demonstrated. Another real-life application of this methodology in bioinformatics is described in Chap. 7 where the problem of selecting discriminative microRNAs from microarray expression data sets is addressed. In the next chapter, another important task of bioinformatics, namely, selection of discriminative genes from microarray gene expression, is handled, where different f-information measures are used as the evaluation criteria for gene selection problem.

References

1. Amat L, Besalu E, Carbo-Dorca R (2001) Identification of active molecular sites using quantum-self-similarity matrices. J Chem Inf Comput Sci 41:978–991
2. Bajorath J, Klein TE, Lybrand TP, Novotny J (1999) Computer-aided drug discovery: from target proteins to drug candidates. Proc Pac Symp Biocomput 4:413–414

3. Bazan J, Skowron A, Synak P (1994) Dynamic reducts as a tool for extracting laws from decision tables. In: Ras ZW, Zemankova M (eds) Proceedings of the 8th symposium on methodologies for intelligent systems. Lecture notes in artificial intelligence, vol 869. Springer, New York, pp 346–355
4. Bjorvand AT, Komorowski J (1997) Practical applications of genetic algorithms for efficient reduct computation. In: Proceedings of the 15th IMACS world congress on scientific computation, modeling and applied mathematics, vol 4, pp 601–606
5. Bravi G, Gancia E, Mascagni P, Pegna M, Todeschini R, Zaliani A (1997) MS-WHIM: New 3D theoretical descriptors derived from molecular surface properties: a comparative 3D QSAR study in a series of steroids. J Comput Aided Mol Des 11:79–92
6. Chen H, Zhou J, Xie G (1998) PARM: a genetic algorithm to predict bioactivity. J Chem Inf Comput Sci 38:243–250
7. Chen KH, Ras ZW, Skowron A (1988) Attributes and rough properties in information systems. Int J Approx Reason 2:365–376
8. Chouchoulas A, Shen Q (2001) Rough set-aided keyword reduction for text categorisation. Appl Artif Intell 15(9):843–873
9. Cornelis C, Jensen R, Martin GH, Slezak D (2010) Attribute selection with fuzzy decision reducts. Inf Sci 180:209–224
10. Devijver PA, Kittler J (1982) Pattern recognition: a statistical approach. Prentice Hall, Englewood Cliffs
11. Dubois D, Prade H (1990) Rough fuzzy sets and fuzzy rough sets. Int J Gen Syst 17:191–209
12. Dubois D, Prade H (1992) Putting fuzzy sets and rough sets together. In: Slowiniski R (ed) Intelligent decision support: handbook of applications and advances of rough sets theory. Kluwer, Dordrecht, pp 203–232
13. Guha R, Jurs PC (2004) Development of linear, ensemble, and nonlinear models for the prediction and interpretation of the biological activity of a set of PDGFR inhibitors. J Chem Inf Comput Sci 44:2179–2189
14. Guha R, Jurs PC (2004) Development of QSAR models to predict and interpret the biological activity of artemisinin analogues. J Chem Inf Comput Sci 44:1440–1449
15. Guyon I (2003) Elisseeff: an introduction to variable and feature selection. J Mach Learn Res 3:1157–1182
16. Hu Q, Xie Z, Yu D (2007) Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation. Pattern Recogn 40:3577–3594
17. Hu Q, Yu D, Liu J, Wu C (2008) Neighborhood rough set based heterogeneous feature subset selection. Inf Sci 178:3577–3594
18. Hu Q, Yu D, Xie Z, Liu J (2007) Fuzzy probabilistic approximation spaces and their information measures. IEEE Trans Fuzzy Syst 14(2):191–201
19. Jain AN, Koile K, Chapman D (1994) Compass: predicting biological activities from molecular surface properties. Performance comparisons on a steroid benchmark. J Med Chem 37:2315–2327
20. Jensen R, Shen Q (2004) Fuzzy-rough attribute reduction with application to web categorization. Fuzzy Sets Syst 141:469–485
21. Jensen R, Shen Q (2004) Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approach. IEEE Trans Knowl Data Eng 16(12):1457–1471
22. Jensen R, Shen Q (2007) Fuzzy-rough sets assisted attribute selection. IEEE Trans Fuzzy Syst 15:73–89
23. Jensen R, Shen Q (2009) New approaches to fuzzy-rough feature selection. IEEE Trans Fuzzy Syst 17(4):824–838
24. Katritzky AR, Lobanov V, karelson M (1994) Comprehensive descriptors for structural and statistical analysis version 1.1. University of Florida, Florida
25. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97(1–2):273–324
26. Koller D, Sahami M (1996) Toward optimal feature selection. In: Proceedings of the international conference on machine learning, pp 284–292

27. Komorowski J, Pawlak Z, Polkowski L, Skowron A (1999) Rough sets: a tutorial. In: Pal SK, Skowron A (eds) Rough-fuzzy hybridization: a new trend in decision making. Springer, Singapore, pp 3–98
28. Kumar M, Thurow K, Stoll N, Stoll R (2007) Robust fuzzy mappings for QSAR studies. Eur J Med Chem 42:675–685
29. Leach AR (2001) Molecular modelling: principles and applications, vol 2. Prentice Hall, Reading
30. Leardi R, Gonzalez AL (1998) Genetic algorithms applied to feature selection in PLS regression: How and when to use them. Chemometr Intell Lab Syst 41:195–207
31. Li ZR, Han LY, Xue Y, Yap CW, Li H, Jiang L, Chen YZ (2007) MODEL—molecular descriptor lab: a web-based server for computing structural and physicochemical features of compounds. Biotechnol Bioeng 97:96–389
32. Lin TY (2001) Granulation and nearest neighborhoods: rough set approach. In: Pedrycz W (ed) Granular computing: an emerging paradigm. Physica-Verlag, Heidelberg, pp 125–142
33. Liu SS, Yin CS, Li ZL, Cai SX (2001) QSAR study of steroid benchmark and dipeptides based on MEDV-13. J Chem Inf Comput Sci 41:321–329
34. Maji P (2009) f-Information measures for efficient selection of discriminative genes from microarray data. IEEE Trans Biomed Eng 56(4):1063–1069
35. Maji P, Garai P (2013) On fuzzy-rough attribute selection: criteria of max-dependency, max-relevance, min-redundancy, and max-significance. Appl Soft Comput 13(9):3968–3980
36. Maji P, Pal SK (2010) Feature selection using f-information measures in fuzzy approximation spaces. IEEE Trans Knowl Data Eng 22(6):854–867
37. Maji P, Paul S (2010) Rough sets for selection of molecular descriptors to predict biological activity of molecules. IEEE Trans Syst Man Cybern Part C Appl Rev 40(6):639–648
38. Modrzejewski M (1993) Feature selection using rough sets theory. In: Proceedings of the 11th international conference on machine learning, pp 213–226
39. Neagu CDN, Aptula AO, Gini G (2002) Neural and neuro-fuzzy models of toxic action of phenols. In: Proceedings of the 1st international IEEE symposium on intelligent systems, vol 1, pp 283–288
40. Ozdemir M, Embrechts MJ, Arciniegas F, Breneman CM, Lockwood L, Bennett KP (2001) Feature selection for in-silico drug design using genetic algorithms and neural networks. In: Proceedings of IEEE mountain workshop on soft computing in industrial applications, pp 25–27
41. Parthalain N, Shen Q, Jensen R (2010) A distance measure approach to exploring the rough set boundary region for attribute reduction. IEEE Trans Knowl Data Eng 22(3):305–317
42. Pawlak Z (1991) Rough sets: theoretical aspects of resoning about data. Kluwer, Dordrecht
43. Polanski J, Walczak B (2000) The comparative molecular surface analysis (COMSA): a novel tool for molecular design. Comput Chem 24:615–625
44. Quinlan JR (1986) Induction of decision trees. Mach Learn 1:81–106
45. Quinlan JR (1993) C4.5: programs for machine learning. Morgan Kaufmann, Mountain View
46. Quinlan JR (1996) Improved use of continuous attributes in C4.5. J Artif Intell Res 4:77–90
47. Robert D, Amat L, Carbo-Dorca R (1999) Three-dimensional quantitative structure-activity relationships from tuned molecular quantum similarity measures: prediction of the corticosteroid-binding globulin binding affinity for a steroid family. J Chem Inf Comput Sci 39:333–344
48. Robinson D, Winn P, Lyne P, Richards W (1999) Self-organizing molecular field analysis: a tool for structure-activity studies. J Med Chem 42:573–583
49. Shen Q, Chouchoulas A (1999) Combining rough sets and data-driven fuzzy learning for generation of classification rules. Pattern Recogn 32(12):2073–2076
50. Skowron A, Rauszer C (1992) The discernibility matrices and functions in information systems. In: Slowinski R (ed) Intelligent decision support. Kluwer, Dordrecht, pp 331–362
51. Skowron A, Swiniarski RW, Synak P (2005) Approximation spaces and information granulation. LNCS Trans Rough Sets 3:175–189

52. Slezak D (1996) Approximate reducts in decision tables. In: Proceedings of the 6th international conference on information processing and management of uncertainty in knowledge-based systems, pp 1159–1164
53. Sventik V, Wang T, Tong C, Liaw A, Sheridan RP, Song Q (2005) Boosting: an ensemble learning tool for compound classification and QSAR modeling. J Chem Inf Model 45(3):786–799
54. Tetkoa IV, Gasteiger J, Todeschini R, Mauri A, Livingstone D, Ertl P, Palyulin VA, Radchenko EV, Zefirov NS, Makarenko AS, Tanchuk VY, Prokopenko VV (2005) Virtual computational chemistry laboratory design and description. J Comput Aided Mol Des 19(6):453–463
55. Tsang ECC, Chen D, Yeung DS, Wang XZ, Lee J (2008) Attributes reduction using fuzzy rough sets. IEEE Trans Fuzzy Syst 16(5):1130–1141
56. Tuppurainen K, Viisas M, Laatikainen R, Peräkylä M (2002) Evaluation of a novel electronic eigenvalue (EEVA) molecular descriptor for QSAR/QSPR studies: validation using a benchmark steroid data set. J Chem Inf Comput Sci 42(3):607–613
57. Turner DB, Willett P, Ferguson AM, Heritage TW (1999) Evaluation of a novel molecular vibration-based descriptor (EVA) for QSAR studies: 2. model validation using a benchmark steroid data set. J Comput Aided Mol Des 13(3):271–296
58. Uddameri V, Kuchanur M (2004) Fuzzy QSARs for predicting log K_{oc} of persistent organic pollutants. Chemosphere 54(6):771–776
59. Vapnik V (1995) The nature of statistical learning theory. Springer-Verlag, New York
60. Wroblewski J (1995) Finding minimal reducts using genetic algorithms. In: Proceedings of the 2nd annual joint conference on information sciences, pp 186–189
61. Wu H, Wu Y, Luo J (2009) An interval type-2 fuzzy rough set model for attribute reduction. IEEE Trans Fuzzy Syst 17(2):301–315
62. Yamaguchi D (2009) Attribute dependency functions considering data efficiency. Int J Approximate Reasoning 51:89–98
63. Zhong N, Dong J, Ohsuga S (2001) Using rough sets with heuristics for feature selection. J Intell Inf Syst 16:199–214
64. Zhou YP, Cai CB, Huan S, Jiang JH, Wu HL, Shen GL, Yu RQ (2007) QSAR study of angiotensin II antagonists using robust boosting partial least squares regression. Anal Chim Acta 593:68–74
65. Ziarko W (1993) Variable precision rough set model. J Comput Syst Sci 46:39–59

Chapter 5
f-Information Measures for Selection of Discriminative Genes from Microarray Data

5.1 Introduction

The wide use of high-throughput technology produces an explosion in using gene expression phenotype for identification and classification in a variety of diagnostic areas. An important application of gene expression data in functional genomics is to classify samples according to their gene expression profiles such as to classify cancer versus normal samples or to classify different types or subtypes of cancer [11, 17, 46].

A microarray gene expression data set can be represented by an expression table, $\mathscr{T} = \{w_{ij} | i = 1, \ldots, m, j = 1, \ldots, n\}$, where $w_{ij} \in \Re$ is the measured expression level of gene $\mathscr{A}_i$ in the jth sample, m and n represent the total number of genes and samples, respectively. Each row in the expression table corresponds to one particular gene and each column to a sample [17]. However, for most gene expression data, the number of training samples is still very small compared to the large number of genes involved in the experiments [17]. For example, the colon cancer data set consists of 62 samples and 2,000 genes and the leukemia data set contains 72 samples and 7,129 genes. The number of samples is likely to remain small for many areas of investigation, especially for human data, due to the difficulty of collecting and processing microarray samples [17]. When the number of genes is significantly greater than the number of samples, it is possible to find biologically relevant correlations of gene behavior with the sample categories [37].

However, among the large amount of genes, only a small fraction is effective for performing a certain task. Also, a small subset of genes is desirable in developing gene expression-based diagnostic tools for delivering precise, reliable, and interpretable results. With the gene selection results, the cost of biological experiment and decision can be greatly reduced by analyzing only the marker genes. Hence, identifying a reduced set of most relevant genes is the goal of gene selection. The small number of training samples and a large number of genes make gene selection a more relevant and challenging problem in gene expression-based classification. This is an important problem in machine learning and referred to as feature selection [9, 31].

P. Maji and S. Paul, *Scalable Pattern Recognition Algorithms*,
DOI: 10.1007/978-3-319-05630-2_5,

In this regard, different feature selection methods [4, 9, 10, 31, 32, 34, 39, 55, 60] can be used to select discriminative genes from microarray data sets. A detailed survey on different feature selection algorithms is reported in Chap. 4. There are also lots of gene selection algorithms developed to select differentially expressed genes [58]. One of the popular gene selection method is significance analysis of microarrays [64], which assigns a score to each gene on the basis of change in gene expression relative to the standard deviation of repeated measurements. Other notable gene selection algorithms are reported in [38, 40, 48, 52, 59, 72].

Due to the high dimensionality of microarray data set, fast, scalable, and efficient feature selection techniques such as univariate filter methods [3, 13, 25, 33, 36, 62] have attracted most attention. Univariate methods can be both parametric [2, 15, 49, 63] and non-parametric [14, 41, 51, 53, 54, 64]. The simplicity of the univariate techniques has made it dominant in the field of gene selection using microarray data. However, the univariate selection methods have certain restrictions and may lead to less accurate classifiers as they do not take into account the gene-gene interactions. Also, the gene sets obtained by these methods contain redundant or similar genes.

The application of multivariate filter methods ranges from simple bivariate interactions [5] to more advanced solutions exploring higher order interactions such as correlation-based feature selection [20, 61, 68, 73] and several variants of the Markov blanket filter method [16, 45, 70]. There also exist a number of feature selection algorithms that group correlated features to reduce the redundancy among the selected features [7, 8, 20, 21, 26, 30, 47]. The uncorrelated shrunken centroid [74] and minimum redundancy-maximum relevance (mRMR) [10, 55] algorithms are two important multivariate filter procedures, highlighting the advantage of using multivariate methods over univariate procedures in the gene expression domain. The mRMR method selects a subset of genes from the whole gene set by maximizing the relevance and minimizing the redundancy of the selected genes. An f-information measure-based method has been reported in [43] for selection of discriminative genes from microarray data using the mRMR criterion. In this regard, it should be noted that the mRMR criterion is also used in [23] and [44] for gene selection, based on the concepts of neighborhood mutual information and fuzzy-rough sets, respectively.

Gene selection using wrapper or embedded methods offers an alternative way to perform a multivariate gene subset selection, incorporating the classifiers; bias into the search and thus offering an opportunity to construct more accurate classifiers. In the context of microarray analysis, most wrapper methods use population-based, randomized search heuristics [4, 29, 35, 50], although some methods use sequential search techniques [24, 71]. An interesting hybrid filter-wrapper approach is introduced in [57], integrating a univariately preordered gene ranking with an incrementally augmenting wrapper method. The embedded capacity of several classifiers to discard input features and thus propose a subset of discriminative genes, has been exploited by several authors. Examples include the use of random forests, a classifier that combines many single decision trees, in an embedded way to calculate the importance of each gene [6, 28, 65]. Another line of embedded feature selection techniques uses the weights of each feature in linear classifiers such as support vector machine [19] and logistic regression [42]. These weights are used to reflect the

relevance of each gene in a multivariate way, and thus allow for the removal of genes with very small weights.

In gene selection process, an optimal gene subset is always relative to a certain criterion. In general, different criteria may lead to different optimal gene subsets. However, every criterion tries to measure the discriminating ability of a gene or a subset of genes to distinguish different class labels. To measure the gene-class relevance, different statistical and information theoretic measures such as the F-test, t-test [10, 34], entropy, information gain, mutual information [10, 55], normalized mutual information [39], and f-information measures [43] are typically used, and the same or a different metric-like mutual information, f-information, the L_1 distance, Euclidean distance, and Pearson's correlation coefficient [10, 27, 55] is employed to calculate the gene-gene redundancy. However, as the F-test, t-test, Euclidean distance, and Pearson's correlation depend on the actual gene expression values of the microarray data, they are very much sensitive to noise or outlier of the data set [10, 22, 27, 55]. On the other hand, as information measures depend only on the probability distribution of a random variable rather than on its actual values, they are more effective to evaluate both gene-class relevance and gene-gene redundancy [18, 39, 55].

However, measures of the distance between a joint probability distribution and product of the marginal distributions are information measures [43, 56, 66]. Information measures constitute a subclass of the divergence measures, which are measures of the distance between two arbitrary distributions. A specific class of information (respectively, divergence) measures, of which mutual information is a member, is formed by the f-information (respectively, f-divergence) measures [43, 56, 66]. In this chapter, several f-information measures are compared with mutual information by applying them to the selection of genes from microarray data. The performance of different information measures is studied using the predictive accuracy of naive Bayes classifier, K-nearest neighbor rule, and support vector machine. The effectiveness of different f-information measures, along with a comparison with mutual information, is demonstrated on three cancer microarray data sets, namely, breast cancer, leukemia, and colon cancer data sets.

The structure of the rest of this chapter is as follows: The problem of gene selection from microarray data sets using several information theoretic measures is described in Sect. 5.2, along with a brief description of different f-information measures. A few case studies and a comparison among different f-information measures are reported in Sect. 5.3. Concluding remarks are given in Sect. 5.4.

5.2 Gene Selection Using f-Information Measures

In microarray data analysis, the data set may contain a number of redundant genes with low relevance to the classes. The presence of such redundant and nonrelevant genes leads to a reduction in the useful information. Ideally, the selected genes should have high relevance with the classes while the redundancy among them should be

as low as possible. The genes with high relevance are expected to be able to predict the classes of the samples. However, the prediction capability is reduced if many redundant genes are selected. In contrast, a data set that contains genes not only with high relevance with respect to the classes but with low mutual redundancy is more effective in its prediction capability. Hence, to assess the effectiveness of the genes, both relevance and redundancy need to be measured quantitatively. In this chapter, the minimum redundancy-maximum relevance framework of Ding and Peng [10, 55] is used to select a set of relevant and nonredundant genes from microarray gene expression data sets.

5.2.1 Minimum Redundancy-Maximum Relevance Criterion

Let $\mathbb{C} = \{\mathscr{A}_1, \ldots, \mathscr{A}_i, \ldots, \mathscr{A}_j, \ldots, \mathscr{A}_m\}$ be the set of m genes of a given microarray gene expression data set and $\mathbb{S}$ is the set of selected genes. Define $\hat{f}(\mathscr{A}_i, \mathbb{D})$ as the relevance of the gene $\mathscr{A}_i$ with respect to the class label $\mathbb{D}$ while $\tilde{f}(\mathscr{A}_i, \mathscr{A}_j)$ as the redundancy between two genes $\mathscr{A}_i$ and $\mathscr{A}_j$. The total relevance of all selected genes is, therefore, given by

$$\mathscr{J}_{\text{relev}} = \sum_{\mathscr{A}_i \in \mathbb{S}} \hat{f}(\mathscr{A}_i, \mathbb{D}) \tag{5.1}$$

while the total redundancy among the selected genes is

$$\mathscr{J}_{\text{redun}} = \sum_{\mathscr{A}_i, \mathscr{A}_j \in \mathbb{S}} \tilde{f}(\mathscr{A}_i, \mathscr{A}_j). \tag{5.2}$$

Therefore, the problem of selecting a set $\mathbb{S}$ of relevant and nonredundant genes from the whole set $\mathbb{C}$ of m genes is equivalent to maximize $\mathscr{J}_{\text{relev}}$ and minimize $\mathscr{J}_{\text{redun}}$, that is, to maximize the objective function $\mathscr{J}$, where

$$\mathscr{J} = \mathscr{J}_{\text{relev}} - \mathscr{J}_{\text{redun}}; \tag{5.3}$$

$$\text{that is,} \quad = \sum_i \hat{f}(\mathscr{A}_i, \mathbb{D}) - \sum_{i,j} \tilde{f}(\mathscr{A}_i, \mathscr{A}_j). \tag{5.4}$$

To solve the above problem, a greedy algorithm is widely used that follows next [10, 55]:

1. Initialize $\mathbb{C} \leftarrow \{\mathscr{A}_1, \ldots, \mathscr{A}_i, \ldots, \mathscr{A}_j, \ldots, \mathscr{A}_m\}$, $\mathbb{S} \leftarrow \emptyset$.
2. Calculate the relevance $\hat{f}(\mathscr{A}_i, \mathbb{D})$ of each gene $\mathscr{A}_i \in \mathbb{C}$.
3. Select gene $\mathscr{A}_i$ as the most relevant gene that has the highest relevance $\hat{f}(\mathscr{A}_i, \mathbb{D})$. In effect, $\mathscr{A}_i \in \mathbb{S}$ and $\mathbb{C} = \mathbb{C} \setminus \mathscr{A}_i$.
4. Repeat the following two steps until the desired number of genes is selected.

5. Calculate the redundancy between already selected genes of $\mathbb{S}$ and each of the remaining genes of $\mathbb{C}$.
6. From the remaining genes of $\mathbb{C}$, select gene $\mathscr{A}_j$ that maximizes

$$\hat{f}(\mathscr{A}_j, \mathbb{D}) - \frac{1}{|\mathbb{S}|} \sum_{\mathscr{A}_i \in \mathbb{S}} \tilde{f}(\mathscr{A}_i, \mathscr{A}_j). \tag{5.5}$$

   As a result of that, $\mathscr{A}_j \in \mathbb{S}$ and $\mathbb{C} = \mathbb{C} \setminus \mathscr{A}_j$.
7. Stop.

5.2.2 *f*-Information Measures for Gene Selection

In this chapter, different f-information measures are reported to compute both gene-class relevance and gene-gene redundancy for selection of genes from microarray data. The f-information measures calculate the distance between a given joint probability p_{ij} and the joint probability when the variables are independent $p_i p_j$. In the following analysis, it is assumed that all probability distributions are complete, that is, $\sum_i p_i = \sum_j p_j = \sum_{i,j} p_{ij} = 1$.

The extent to which two probability distributions differ can be expressed by a so-called measure of divergence. Such a measure will reach a minimum value when the two probability distributions are identical and the value increases with increasing disparity between the two distributions. A specific class of divergence measures is the set of f-divergence measures [56, 66]. For two discrete probability distributions $P = \{p_i |\ i = 1, 2, \ldots, n\}$ and $Q = \{q_i |\ i = 1, 2, \ldots, n\}$, the f-divergence is defined as

$$f(P||Q) = \sum_i q_i f\left(\frac{p_i}{q_i}\right). \tag{5.6}$$

The demands on the function f are that

1. $f : [0, \infty) \to (-\infty, \infty]$;
2. f is continuous and convex on $[0, \infty)$;
3. finite on $(0, \infty)$; and
4. strictly convex at some point $x \in (0, \infty)$.

The following definition completes the definition of f-divergence for the two cases for which (5.6) is not defined:

$$q_i f\left(\frac{p_i}{q_i}\right) = \begin{cases} 0, & \text{if } p_i = q_i = 0 \\ p_i \lim\limits_{x \to \infty} \dfrac{f(x)}{x}, & \text{if } p_i > 0, q_i = 0. \end{cases} \tag{5.7}$$

A special case of f-divergence measures is the f-information measures. These are defined similarly to f-divergence measures, but apply only to specific probability distributions; namely, the joint probability of two variables P and their marginal probabilities' product $P_1 \times P_2$. Thus, the f-information is a measure of dependence: it measures the distance between a given joint probability and the joint probability when the variables are independent [56, 66]. The frequently used functions that can be used to form f-information measures include V-information, I_α-information, M_α-information, and χ^α-information. On the other hand, the Renyi's distance measure does not fall in the class of f-divergence measures as it does not satisfy the definition of f-divergence. However, it is divergence measure in the sense that it measures the distance between two distributions and it is directly related to f-divergence.

5.2.2.1 V-Information

One of the simplest measures of dependence can be obtained using the function $V = |x - 1|$, which results in the V-information [56, 66]

$$V(P||P_1 \times P_2) = \sum_{i,j} |p_{ij} - p_i p_j| \tag{5.8}$$

where $P_1 = \{p_i |\ i = 1, 2, \ldots, n\}$, $P_2 = \{p_j |\ j = 1, 2, \ldots, n\}$, and $P = \{p_{ij} |\ i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, n\}$ represent two marginal probability distributions and their joint probability distribution, respectively. Hence, the V-information calculates the absolute distance between joint probability of two variables and their marginal probabilities' product.

5.2.2.2 I_α-Information

The I_α-information is defined as [56, 66]

$$I_\alpha(P||P_1 \times P_2) = \frac{1}{\alpha(\alpha - 1)} \left(\sum_{i,j} \frac{(p_{ij})^\alpha}{(p_i p_j)^{\alpha - 1}} - 1 \right) \tag{5.9}$$

for $\alpha \neq 0, \alpha \neq 1$. The class of I_α-information includes mutual information, which equals I_α for the limit $\alpha \to 1$. That is,

$$I_1(P||P_1 \times P_2) = \sum_{i,j} p_{ij} \log \left(\frac{p_{ij}}{p_i p_j} \right) \text{ for } \alpha \to 1. \tag{5.10}$$

5.2.2.3 M_α-Information

The M_α-information, defined by Matusita [56, 66], is as follows:

$$M_\alpha(x) = |x^\alpha - 1|^{\frac{1}{\alpha}}, \quad 0 < \alpha \leq 1. \tag{5.11}$$

When applying this function in the definition of an f-information measure, the resulting M_α-information measures are

$$M_\alpha(P||P_1 \times P_2) = \sum_{i,j} |(p_{ij})^\alpha - (p_i p_j)^\alpha|^{\frac{1}{\alpha}} \tag{5.12}$$

for $0 < \alpha \leq 1$. These constitute a generalized version of V-information. That is, the M_α-information is identical to V-information for $\alpha = 1$.

5.2.2.4 χ^α-Information

The class of χ^α-information measures, proposed by Liese and Vajda [66], is as follows:

$$\chi^\alpha(x) = \begin{cases} |1 - x^\alpha|^{\frac{1}{\alpha}} & \text{for } 0 < \alpha \leq 1 \\ |1 - x|^\alpha & \text{for } \alpha > 1. \end{cases} \tag{5.13}$$

For $0 < \alpha \leq 1$, this function equals to the M_α function. The χ^α-information and M_α-information measures are, therefore, also identical for $0 < \alpha \leq 1$. For $\alpha > 1$, χ^α-information can be written as

$$\chi^\alpha(P||P_1 \times P_2) = \sum_{i,j} \frac{|p_{ij} - p_i p_j|^\alpha}{(p_i p_j)^{\alpha-1}}. \tag{5.14}$$

5.2.2.5 Renyi Distance

The Renyi distance, a measure of information of order α [56, 66], can be defined as

$$R_\alpha(P||P_1 \times P_2) = \frac{1}{\alpha - 1} \log \sum_{i,j} \frac{(p_{ij})^\alpha}{(p_i p_j)^{\alpha-1}}$$

for $\alpha \neq 0, \alpha \neq 1$. It reaches its minimum value when p_{ij} and $p_i p_j$ are identical, in which case the summation reduces to $\sum p_{ij}$. As complete probability distribution is assumed, the sum is one and the minimum value of the measure is, therefore, equal to zero. The limit of Renyi's measure for α approaching 1 equals $I_1(P||P_1 \times P_2)$, which is the mutual information.

5.2.3 Discretization

In microarray gene expression data sets, the class labels of samples are represented by discrete symbols, while the expression values of genes are continuous. Hence, to measure both gene-class relevance of a gene with respect to class labels and gene-gene redundancy between two genes using information theoretic measures such as mutual information [10, 55], normalized mutual information [39], and f-information measures [43], the continuous expression values of a gene are divided into several discrete partitions. The a prior (marginal) probabilities and their joint probabilities are then calculated to compute both gene-class relevance and gene-gene redundancy using the definitions for discrete cases. In this chapter, the discretization method reported in [10, 43, 55] is employed to discretize the continuous gene expression values. The expression values of a gene are discretized using mean μ and standard deviation σ computed over n expression values of that gene: any value larger than $(\mu + \sigma/2)$ is transformed to state 1; any value between $(\mu - \sigma/2)$ and $(\mu + \sigma/2)$ is transformed to state 0; any value smaller than $(\mu - \sigma/2)$ is transformed to state -1. These three states correspond to the over-expression, baseline, and under-expression of genes.

5.3 Experimental Results

The performance of different f-information measures is extensively compared with that of mutual information and normalized mutual information. Based on the argumentation given in Sect. 5.2.2, the following information measures are chosen to include in the study:

1. I_α- and R_α-information measures for $\alpha \neq 0$ and $\alpha \neq 1$;
2. mutual information ($I_{1.0}$- and $R_{1.0}$-information);
3. M_α-information measure for $0 < \alpha \leq 1$;
4. χ^α-information measure for $\alpha > 1$; and
5. normalized mutual information U.

In this chapter, these measures are applied to calculate both gene-class relevance and gene-gene redundancy. The minimum redundancy-maximum relevance (mRMR) criterion [10, 55] is used for gene selection. The source code of the f-information based mRMR (f-mRMR) algorithm [43], written in C language, is available at http://www.isical.ac.in/~bibl/results/fmRMR/fmRMR.html. All the information measures are implemented in C language and run in LINUX environment having machine configuration Pentium IV, 3.2 GHz, 1 MB cache, and 1 GB RAM.

To analyze the performance of different f-information measures, the experimentation is done on three microarray gene expression data sets. The major metric for evaluating the performance of different measures is the classification accuracy of support vector machine (SVM) [67], K-nearest neighbor (K-NN) rule [12], and naive Bayes (NB) classifier [12].

5.3.1 Gene Expression Data Sets

In this chapter, three public data sets of cancer microarrays are used. Since binary classification is a typical and fundamental issue in diagnostic and prognostic prediction of cancer, different f-information measures are compared using following binary-class data sets.

5.3.1.1 Breast Cancer Data Set

The breast cancer data set contains expression levels of 7,129 genes in 49 breast tumor samples [69]. The samples are classified according to their estrogen receptor (ER) status: 25 samples are ER positive while the other 24 samples are ER negative.

5.3.1.2 Leukemia Data Set

The leukemia data set is an Affymetrix high-density oligonucleotide array that contains 7,070 genes and 72 samples from two classes of leukemia: 47 acute lymphoblastic leukemia and 25 acute myeloid leukemia [17]. The data set is publicly available at http://www.broad.mit.edu/cgibin/cancer/datasets.cgi.

5.3.1.3 Colon Cancer Data Set

The colon cancer data set contains expression levels of 2,000 genes and 62 samples from two classes [1]: 40 tumor and 22 normal colon tissues. The data set is available at http://microarray.princeton.edu/oncology/affydata/index.html.

5.3.2 Class Prediction Methods

The SVM [67], K-NN rule [12], and NB classifier [12] are used to evaluate the performance of different f-information measures. A brief introduction of the SVM is reported in Chaps. 3 and 4. In this work, linear kernels are used in the SVM to construct the nonlinear decision boundary. On the other hand, descriptions of both K-NN rule and NB classifier are reported next.

5.3.2.1 K-Nearest Neighbor Rule

The K-nearest neighbor (K-NN) rule [12] is used for evaluating the effectiveness of the reduced feature set for classification. It classifies samples based on its closest

training samples in the feature space. A sample is classified by a majority vote of its K-neighbors, with the sample being assigned to the class most common amongst its K-nearest neighbors. The value of K, chosen for the K-NN, is the square root of number of samples in training set.

5.3.2.2 Naive Bayes Classifier

The naive Bayes (NB) classifier [12] is one of the oldest classifiers. It is obtained by using the Bayes rule and assuming features or variables are independent of each other given its class. For the jth sample x_j with m gene expression levels $\{w_{1j}, \ldots, w_{ij}, \ldots, w_{mj}\}$ for the m genes, the posterior probability that x_j belongs to class c is

$$p(c|x_j) \propto \prod_{i=1}^{m} p(w_{ij}|c) \tag{5.15}$$

where $p(w_{ij}|c)$ are conditional tables or conditional density estimated from training examples.

5.3.3 Performance Analysis

The experimental results on three microarray data sets are presented in Tables 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8 and 5.9. Subsequent discussions analyze the results with respect to the prediction accuracy of the NB, SVM, and K-NN classifiers. Tables 5.1, 5.2, 5.4, 5.5, 5.7, and 5.8 provide the performance of different f-information measures using the NB and SVM, respectively, while Tables 5.3, 5.6 and 5.9 shows the results using the K-NN rule. The values of α for f-information measures investigated are 0.2, 0.5, 0.8, 1.5, 2.0, 3.0, and 4.0. Some measures resemble mutual information for $\alpha = 1.0$ (I_α and R_α) and some resemble another measure ($M_{1.0}$ and $\chi^{1.0}$ equal V). To compute the prediction accuracy of the NB, SVM, and K-NN, the leave-one-out cross-validation is performed on each gene expression data set. The number of genes selected ranges from 2 to 50 and each data set is preprocessed by standardizing each sample to zero mean and unit variance.

Tables 5.1, 5.4 and 5.7 shows that, for three microarray data sets, genes selected by $I_{0.2}$-, $M_{0.2}$-, and $R_{0.2}$-information measures lead to higher classification accuracy than those selected by mutual information and other f-information measures. With the NB, a classification accuracy of 100 % is obtained for $I_{0.2}$- and $R_{0.2}$-information measures considering 15 or more genes in case of breast cancer data, eight or ten genes in case of leukemia data, and 25 or more genes in case of colon cancer data, while in case of $M_{0.2}$-information measure 15 or more genes for breast cancer data, eight or more genes for leukemia data, and 30 or more genes for colon cancer data are required to achieve this accuracy. Similarly, 100 % accuracy for breast cancer data is obtained

Table 5.1 Performance on breast cancer data set using NB classifier

f-Information measures	Number of selected genes											
	2	5	8	10	15	20	25	30	35	40	45	50
$I_{0.2}$	95.9	98.0	98.0	98.0	100	100	100	100	100	100	98.0	98.0
$I_{0.5}$	95.9	98.0	98.0	98.0	100	100	100	98.0	98.0	98.0	98.0	98.0
$I_{0.8}$	95.9	100	95.9	98.0	98.0	98.0	95.9	93.9	91.8	91.8	89.8	89.8
$I_{1.0}$	95.9	98.0	95.9	100	98.0	93.9	93.9	89.8	87.8	87.8	87.8	87.8
$I_{1.5}$	95.9	98.0	95.9	93.9	93.9	91.8	91.8	89.8	85.7	83.7	83.7	81.6
$I_{2.0}$	95.9	95.9	95.9	93.9	91.8	91.8	91.8	87.8	87.8	83.7	83.7	81.6
$I_{3.0}$	95.9	95.9	95.9	93.9	91.8	91.8	89.8	87.8	87.8	83.7	83.7	81.6
$I_{4.0}$	95.9	95.9	95.9	91.8	91.8	89.8	87.8	87.8	85.7	83.7	81.6	81.6
$M_{0.2}$	85.7	95.9	95.9	98.0	100	100	100	100	98.0	98.0	98.0	98.0
$M_{0.5}$	95.9	98.0	98.0	98.0	100	100	100	98.0	98.0	98.0	98.0	98.0
$M_{0.8}$	95.9	93.9	95.9	98.0	93.9	91.8	91.8	87.8	85.7	85.7	85.7	79.6
$M_{1.0}$	87.8	89.8	83.7	85.7	89.8	87.8	87.8	83.7	85.7	85.7	83.7	83.7
$\chi^{1.5}$	95.9	98.0	95.9	98.0	93.9	89.8	89.8	85.7	83.7	81.6	79.6	79.6
$\chi^{2.0}$	95.9	95.9	95.9	93.9	91.8	91.8	91.8	87.8	87.8	83.7	83.7	81.6
$\chi^{3.0}$	95.9	95.9	95.9	93.9	93.9	93.9	93.9	89.8	87.8	85.7	83.7	83.7
$\chi^{4.0}$	95.9	98.0	100	95.9	95.9	93.9	93.9	89.8	85.7	85.7	85.7	85.7
$R_{0.2}$	95.9	98.0	98.0	98.0	100	100	100	100	100	100	98.0	98.0
$R_{0.5}$	95.9	98.0	98.0	98.0	100	100	100	98.0	98.0	98.0	98.0	98.0
$R_{0.8}$	95.9	100	95.9	95.9	98.0	98.0	95.9	93.9	91.8	91.8	89.8	89.8
$R_{1.0}$	95.9	98.0	95.9	100	98.0	93.9	93.9	89.8	87.8	87.8	87.8	87.8
$R_{1.5}$	95.9	98.0	95.9	93.9	91.8	91.8	91.8	89.8	87.8	83.7	83.7	83.7
$R_{2.0}$	95.9	91.8	95.9	95.9	91.8	91.8	91.8	89.8	85.7	83.7	83.7	81.6
$R_{3.0}$	93.9	89.8	93.9	93.9	93.9	91.8	91.8	91.8	89.8	85.7	83.7	79.6
$R_{4.0}$	93.9	93.9	91.8	91.8	91.8	91.8	89.8	89.8	87.8	83.7	83.7	81.6
U	95.9	98.0	98.0	100	95.9	93.9	93.9	91.8	91.8	89.8	89.8	89.8

for $I_{0.8}$- and $R_{0.8}$-information measures using only five genes. However, both mutual information and normalized mutual information provide maximum 98.6 % accuracy for leukemia data, 93.6 and 95.2 % accuracy for colon cancer data, and 100 % for breast cancer data using 10 genes.

The results reported in Tables 5.2, 5.5 and 5.8 are based on the predictive accuracy of the SVM. The results show that in case of breast cancer data set, the f-information measures along with mutual information and normalized mutual information achieve 100 % classification accuracy. While at least 8 genes are required for mutual information and normalized mutual information to attain this accuracy, $I_{0.2}$-, $I_{0.5}$-, $M_{0.5}$-, $R_{0.2}$-, and $R_{0.5}$-information measures need only 5 genes. On the other hand, both mutual information and normalized mutual information provide maximum 98.6 % accuracy for leukemia data using 25 genes, while $I_{1.5}$-, $M_{0.8}$-, and V- (that is, $M_{1.0}$-) information measures give 100 % accuracy using only 15, 15, and 20 genes, respectively. Similarly, for colon cancer data set, while mutual information and normalized mutual information attain maximum 88.7 and 85.5 % accuracy, respectively, $M_{1.0}$-,

Table 5.2 Performance on breast cancer data set using SVM

f-Information measures	Number of selected genes											
	2	5	8	10	15	20	25	30	35	40	45	50
$I_{0.2}$	81.6	100	95.9	98.0	98.0	100	95.9	95.9	98.0	98.0	98.0	95.9
$I_{0.5}$	81.6	100	100	100	95.9	95.9	100	95.9	95.9	95.9	98.0	98.0
$I_{0.8}$	81.6	98.0	100	100	98.0	95.9	95.9	98.0	98.0	95.9	98.0	95.9
$I_{1.0}$	81.6	98.0	100	100	98.0	95.9	95.9	93.9	93.9	93.9	95.9	95.9
$I_{1.5}$	85.7	91.8	98.0	100	98.0	100	95.9	95.9	95.9	95.9	93.9	93.9
$I_{2.0}$	85.7	95.9	98.0	100	100	100	95.9	95.9	95.9	93.9	93.9	93.9
$I_{3.0}$	85.7	95.9	98.0	100	100	95.9	95.9	95.9	95.9	95.9	93.9	93.9
$I_{4.0}$	85.7	89.8	100	98.0	100	95.9	95.9	95.9	95.9	95.9	95.9	95.9
$M_{0.2}$	77.6	95.9	91.8	89.8	87.8	93.9	93.9	95.9	95.9	95.9	95.9	98.0
$M_{0.5}$	81.6	100	100	100	95.9	95.9	100	95.9	95.9	95.9	98.0	98.0
$M_{0.8}$	85.7	89.8	93.9	89.8	93.9	95.9	93.9	93.9	93.9	91.8	93.9	93.9
$M_{1.0}$	83.7	81.6	87.8	91.8	87.8	83.7	83.7	83.7	85.7	83.7	87.8	85.7
$\chi^{1.5}$	85.7	87.8	91.8	89.8	93.9	91.8	95.9	95.9	93.9	93.9	93.9	93.9
$\chi^{2.0}$	85.7	95.9	98.0	100	100	100	95.9	95.9	95.9	93.9	93.9	93.9
$\chi^{3.0}$	85.7	89.8	100	95.9	98.0	95.9	98.0	93.9	93.9	93.9	93.9	93.9
$\chi^{4.0}$	85.7	91.8	100	100	98.0	95.9	95.9	95.9	95.9	95.9	95.9	95.9
$R_{0.2}$	81.6	100	95.9	98.0	98.0	98.0	95.9	95.9	95.9	98.0	98.0	98.0
$R_{0.5}$	81.6	100	100	100	95.9	95.9	100	95.9	95.9	93.9	98.0	98.0
$R_{0.8}$	81.6	98.0	100	100	98.0	95.9	95.9	98.0	98.0	95.9	98.0	95.9
$R_{1.0}$	81.6	98.0	100	100	98.0	95.9	95.9	93.9	93.9	93.9	95.9	95.9
$R_{1.5}$	85.7	91.8	98.0	100	98.0	100	95.9	95.9	95.9	95.9	93.9	93.9
$R_{2.0}$	85.7	89.8	95.9	95.9	98.0	100	95.9	95.9	95.9	95.9	93.9	93.9
$R_{3.0}$	87.8	87.8	100	100	93.9	95.9	93.9	95.9	95.9	95.9	95.9	95.9
$R_{4.0}$	87.8	89.8	89.8	93.9	98.0	100	100	98.0	98.0	95.9	95.9	93.9
U	81.6	98.0	100	100	98.0	95.9	98.0	95.9	95.9	95.9	95.9	93.9

$\chi^{1.5}$-, and $R_{2.0}$-information measures provide maximum 91.9 % accuracy, and both $I_{2.0}$- and $\chi^{2.0}$-information measures provide maximum 90.3 % accuracy.

For breast cancer data set using the K-NN, 100 % accuracy is obtained in case of mutual information as well as I_α- ($\alpha = 1.5, 2.0, 3.0$), χ^α- ($\alpha = 2.0, 3.0, 4.0$), and R_α- ($\alpha = 1.5, 3.0, 4.0$) information measures, although normalized mutual information provides maximum 98.0 % accuracy. For the K-NN, while mutual information and normalized mutual information achieve maximum 97.2 and 98.6 % accuracy using at least 15 and 30 genes in case of leukemia data set, the V- or $M_{1.0}$-information measure provides 98.6 % accuracy using only eight genes. Similarly, the $I_{4.0}$- and $\chi^{3.0}$-information measures achieve 90.3 % predictive accuracy for colon cancer data set while mutual information and normalized mutual information provide maximum 88.7 % accuracy. However, in case of the K-NN-based results for both leukemia and colon cancer data sets, the majority of f-information measures produces results similar to those of mutual information and normalized mutual information.

Table 5.3 Performance on breast cancer data set using K-NN rule

f-Information measures	Number of selected genes											
	2	5	8	10	15	20	25	30	35	40	45	50
$I_{0.2}$	89.8	93.9	93.9	95.9	98.0	95.9	95.9	93.9	95.9	98.0	98.0	98.0
$I_{0.5}$	89.8	93.9	95.9	95.9	98.0	98.0	95.9	95.9	98.0	98.0	95.9	98.0
$I_{0.8}$	89.8	98.0	95.9	95.9	98.0	95.9	95.9	98.0	98.0	98.0	98.0	98.0
$I_{1.0}$	89.8	98.0	100	98.0	100	95.9	93.9	98.0	95.9	95.9	98.0	98.0
$I_{1.5}$	85.7	93.9	100	100	98.0	95.9	93.9	95.9	98.0	95.9	95.9	95.9
$I_{2.0}$	85.7	91.8	100	100	98.0	95.9	95.9	98.0	98.0	98.0	95.9	95.9
$I_{3.0}$	85.7	91.8	100	100	98.0	93.9	95.9	98.0	98.0	95.9	95.9	95.9
$I_{4.0}$	85.7	91.8	95.9	98.0	98.0	95.9	98.0	95.9	95.9	95.9	98.0	93.9
$M_{0.2}$	83.7	93.9	85.7	83.7	87.8	89.8	85.7	85.7	85.7	87.8	89.8	89.8
$M_{0.5}$	89.8	93.9	95.9	95.9	98.0	98.0	95.9	95.9	98.0	98.0	95.9	98.0
$M_{0.8}$	85.7	89.8	85.7	91.8	95.9	95.9	95.9	93.9	93.9	93.9	93.9	93.9
$M_{1.0}$	71.4	89.8	89.8	89.8	89.8	91.8	91.8	91.8	93.9	93.9	91.8	91.8
$\chi^{1.5}$	85.7	89.8	93.9	91.8	95.9	93.9	95.9	93.9	93.9	93.9	93.9	93.9
$\chi^{2.0}$	85.7	91.8	100	100	98.0	95.9	95.9	98.0	98.0	98.0	95.9	95.9
$\chi^{3.0}$	85.7	91.8	95.9	98.0	98.0	98.0	100	98.0	98.0	98.0	98.0	98.0
$\chi^{4.0}$	85.7	98.0	100	100	98.0	98.0	100	100	98.0	98.0	98.0	98.0
$R_{0.2}$	89.8	93.9	93.9	95.9	98.0	98.0	95.9	93.9	91.8	98.0	98.0	98.0
$R_{0.5}$	89.8	93.9	95.9	95.9	98.0	98.0	95.9	95.9	98.0	98.0	95.9	98.0
$R_{0.8}$	89.8	98.0	95.9	95.9	98.0	95.9	95.9	98.0	98.0	98.0	98.0	98.0
$R_{1.0}$	89.8	98.0	100	98.0	100	95.9	93.9	98.0	95.9	95.9	98.0	98.0
$R_{1.5}$	85.7	93.9	100	100	95.9	95.9	95.9	93.9	95.9	95.9	98.0	93.9
$R_{2.0}$	85.7	91.8	98.0	95.9	95.9	95.9	95.9	93.9	95.9	95.9	95.9	98.0
$R_{3.0}$	91.8	98.0	100	95.9	93.9	95.9	98.0	100	100	98.0	98.0	95.9
$R_{4.0}$	91.8	89.8	95.9	95.9	93.9	95.9	100	100	98.0	100	100	98.0
U	89.8	98.0	98.0	98.0	98.0	95.9	93.9	93.9	95.9	93.9	93.9	93.9

From the results reported here, it is seen that, for a particular number of selected genes, the predictive accuracy for some f-information measures is higher compared to that of mutual information and normalized mutual information, irrespective of the classification models and microarray data sets used. Also, the I_α-, M_α-, and R_α-information measures attain 100 % prediction accuracy using the NB for $\alpha = 0.2$ in all three data sets. In all cases, the $M_{0.5}$- and $I_{0.5}$-information measures provide same results as well as the $\chi^{2.0}$- and $I_{2.0}$-information measures show exactly same performance as they are related by the following relations:

$$I_{0.5}(P||P_1 \times P_2) = 2M_{0.5}(P||P_1 \times P_2) \tag{5.16}$$

$$\chi^{2.0}(P||P_1 \times P_2) = 2I_{2.0}(P||P_1 \times P_2). \tag{5.17}$$

For both colon cancer data set and leukemia data set, 50 top-ranked genes selected by $I_{0.2}$-, $M_{0.2}$- and $R_{0.2}$-information measures based on mRMR criterion are available at http://www.isical.ac.in/~bibl/results/fmRMR/fmRMR.html.

Table 5.4 Performance on leukemia data set using NB classifier

f-Information measures	Number of selected genes											
	2	5	8	10	15	20	25	30	35	40	45	50
$I_{0.2}$	97.2	98.6	100	100	97.2	97.2	95.8	95.8	95.8	94.4	94.4	93.1
$I_{0.5}$	98.6	97.2	97.2	95.8	95.8	95.8	94.4	93.1	93.1	93.1	93.1	90.3
$I_{0.8}$	98.6	98.6	97.2	95.8	94.4	93.1	90.3	87.5	87.5	86.1	86.1	84.7
$I_{1.0}$	98.6	98.6	95.8	95.8	94.4	90.3	87.5	86.1	86.1	86.1	84.7	84.7
$I_{1.5}$	94.4	97.2	95.8	95.8	91.7	87.5	84.7	84.7	84.7	84.7	84.7	84.7
$I_{2.0}$	94.4	97.2	95.8	95.8	91.7	88.9	84.7	84.7	84.7	84.7	84.7	84.7
$I_{3.0}$	94.4	97.2	97.2	94.4	94.4	87.5	84.7	83.3	84.7	84.7	84.7	84.7
$I_{4.0}$	94.4	95.8	97.2	95.8	90.3	86.1	83.3	83.3	83.3	84.7	84.7	84.7
$M_{0.2}$	87.5	95.8	100	100	100	100	100	100	100	100	100	98.6
$M_{0.5}$	98.6	97.2	97.2	95.8	95.8	95.8	94.4	93.1	93.1	93.1	93.1	90.3
$M_{0.8}$	100	97.2	97.2	95.8	91.7	88.9	86.1	86.1	84.7	84.7	83.3	83.3
$M_{1.0}$	94.4	95.8	94.4	94.4	88.9	90.3	87.5	84.7	83.3	81.9	81.9	81.9
$\chi^{1.5}$	94.4	97.2	97.2	95.8	90.3	87.5	86.1	86.1	84.7	84.7	84.7	84.7
$\chi^{2.0}$	94.4	97.2	95.8	95.8	91.7	88.9	84.7	84.7	84.7	84.7	84.7	84.7
$\chi^{3.0}$	94.4	97.2	97.2	94.4	90.3	87.5	84.7	84.7	84.7	84.7	84.7	84.7
$\chi^{4.0}$	95.8	98.6	95.8	94.4	91.7	88.9	86.1	86.1	84.7	84.7	84.7	84.7
$R_{0.2}$	97.2	98.6	100	100	98.6	97.2	95.8	95.8	94.4	94.4	93.1	93.1
$R_{0.5}$	98.6	97.2	97.2	95.8	95.8	95.8	94.4	93.1	93.1	93.1	93.1	90.3
$R_{0.8}$	98.6	97.2	95.8	95.8	94.4	91.7	90.3	87.5	87.5	86.1	86.1	86.1
$R_{1.0}$	98.6	98.6	95.8	95.8	94.4	90.3	87.5	86.1	86.1	86.1	84.7	84.7
$R_{1.5}$	98.6	98.6	97.2	97.2	93.1	88.9	86.1	86.1	84.7	84.7	84.7	84.7
$R_{2.0}$	98.6	97.2	95.8	94.4	93.1	88.9	86.1	84.7	84.7	84.7	84.7	84.7
$R_{3.0}$	98.6	97.2	95.8	94.4	91.7	88.9	87.5	84.7	84.7	84.7	84.7	84.7
$R_{4.0}$	98.6	97.2	97.2	94.4	93.1	91.7	88.9	87.5	86.1	84.7	84.7	84.7
U	98.6	97.2	95.8	94.4	93.1	90.3	88.9	87.5	86.1	86.1	86.1	84.7

5.3.4 Analysis Using Class Separability Index

In case of leukemia and colon cancer data sets, I_α-, M_α- and R_α-information measures provide significantly better results for $\alpha = 0.2$ compared to mutual information and normalized mutual information. In order to analyze the results of these measures further, the class separability index is used next. The class separability index $\mathscr{S}$ [9] of a data set is defined as follows:

$$\mathscr{S} = \text{trace}(V_B^{-1} V_W), \tag{5.18}$$

where V_W is the within class scatter matrix and V_B is the between class scatter matrix, defined as follows:

$$V_W = \sum_{j=1}^{C} \pi_j E\{(X - \mu_j)(X - \mu_j)^T | c_j\} = \sum_{j=1}^{C} \pi_j \Sigma_j; \tag{5.19}$$

Table 5.5 Performance on leukemia data set using SVM

f-Information measures	Number of selected genes											
	2	5	8	10	15	20	25	30	35	40	45	50
$I_{0.2}$	94.4	94.4	94.4	94.4	93.1	94.4	97.2	97.2	97.2	95.8	94.4	97.2
$I_{0.5}$	94.4	95.8	95.8	95.8	97.2	95.8	95.8	97.2	97.2	98.6	98.6	98.6
$I_{0.8}$	94.4	94.4	95.8	95.8	95.8	98.6	97.2	98.6	97.2	97.2	97.2	97.2
$I_{1.0}$	94.4	94.4	93.1	95.8	95.8	98.6	98.6	98.6	98.6	97.2	95.8	97.2
$I_{1.5}$	93.1	97.2	97.2	95.8	100	97.2	97.2	97.2	97.2	98.6	98.6	97.2
$I_{2.0}$	93.1	95.8	95.8	94.4	98.6	97.2	97.2	97.2	98.6	98.6	98.6	97.2
$I_{3.0}$	93.1	95.8	97.2	95.8	97.2	97.2	97.2	95.8	95.8	97.2	97.2	98.6
$I_{4.0}$	93.1	97.2	97.2	97.2	98.6	95.8	97.2	95.8	95.8	97.2	98.6	98.6
$M_{0.2}$	93.1	97.2	97.2	95.8	95.8	94.4	93.1	94.4	94.4	93.1	94.4	98.6
$M_{0.5}$	94.4	95.8	95.8	95.8	97.2	95.8	95.8	97.2	97.2	98.6	98.6	98.6
$M_{0.8}$	90.3	97.2	94.4	95.8	100	98.6	98.6	95.8	95.8	97.2	95.8	95.8
$M_{1.0}$	90.3	97.2	98.6	98.6	98.6	100	97.2	97.2	95.8	97.2	97.2	95.8
$\chi^{1.5}$	93.1	95.8	97.2	98.6	97.2	97.2	95.8	97.2	98.6	97.2	97.2	95.8
$\chi^{2.0}$	93.1	95.8	95.8	94.4	98.6	97.2	97.2	97.2	98.6	98.6	98.6	97.2
$\chi^{3.0}$	93.1	97.2	97.2	95.8	98.6	94.4	98.6	98.6	97.2	97.2	97.2	98.6
$\chi^{4.0}$	93.1	97.2	97.2	97.2	97.2	98.6	98.6	97.2	97.2	97.2	97.2	98.6
$R_{0.2}$	94.4	95.8	94.4	94.4	93.1	94.4	97.2	97.2	97.2	95.8	95.8	95.8
$R_{0.5}$	94.4	95.8	95.8	95.8	97.2	95.8	95.8	97.2	97.2	98.6	98.6	98.6
$R_{0.8}$	94.4	95.8	95.8	95.8	95.8	98.6	97.2	98.6	98.6	97.2	97.2	98.6
$R_{1.0}$	94.4	94.4	93.1	95.8	95.8	98.6	98.6	98.6	98.6	97.2	95.8	97.2
$R_{1.5}$	94.4	94.4	98.6	97.2	97.2	98.6	97.2	97.2	98.6	98.6	98.6	97.2
$R_{2.0}$	94.4	93.1	95.8	94.4	97.2	98.6	97.2	97.2	98.6	98.6	98.6	98.6
$R_{3.0}$	94.4	93.1	95.8	97.2	97.2	98.6	95.8	95.8	94.4	97.2	98.6	98.6
$R_{4.0}$	94.4	94.4	95.8	97.2	97.2	97.2	97.2	97.2	97.2	97.2	97.2	98.6
U	94.4	95.8	95.8	95.8	95.8	95.8	98.6	97.2	98.6	98.6	97.2	95.8

$$V_B = \sum_{j=1}^{C} \pi_j (\mu_j - \bar{\mu})(\mu_j - \bar{\mu})^T; \tag{5.20}$$

$$\text{and } \bar{\mu} = E\{X\} = \sum_{j=1}^{C} \pi_j \mu_j; \tag{5.21}$$

where C is the number of classes, π_j is a priori probability that a pattern belongs to class c_j, X is a feature vector, $\bar{\mu}$ is the sample mean vector for the entire data points, μ_j and Σ_j represent the sample mean and covariance matrix of class c_j, respectively and $E\{\cdot\}$ is the expectation operator. A lower value of $\mathscr{S}$ ensures that classes are well separated by their scatter means.

Figure 5.1 represents the variation of class separability index $\mathscr{S}$ with respect to the number of selected genes for colon cancer data set as an example. For $I_{0.2}$-,

Table 5.6 Performance on leukemia data set using K-NN rule

f-Information measures	Number of selected genes											
	2	5	8	10	15	20	25	30	35	40	45	50
$I_{0.2}$	91.7	94.4	95.8	94.4	97.2	97.2	97.2	97.2	97.2	97.2	97.2	97.2
$I_{0.5}$	94.4	94.4	95.8	95.8	95.8	95.8	97.2	97.2	97.2	97.2	97.2	98.6
$I_{0.8}$	94.4	94.4	94.4	94.4	97.2	97.2	97.2	97.2	97.2	97.2	97.2	94.4
$I_{1.0}$	94.4	94.4	94.4	94.4	97.2	97.2	97.2	95.8	97.2	94.4	94.4	97.2
$I_{1.5}$	93.1	95.8	94.4	95.8	97.2	95.8	97.2	97.2	95.8	97.2	97.2	97.2
$I_{2.0}$	93.1	93.1	95.8	95.8	97.2	98.6	97.2	97.2	97.2	97.2	97.2	97.2
$I_{3.0}$	93.1	93.1	95.8	94.4	97.2	97.2	97.2	97.2	97.2	97.2	97.2	97.2
$I_{4.0}$	93.1	94.4	95.8	97.2	97.2	97.2	97.2	97.2	97.2	97.2	98.6	97.2
$M_{0.2}$	93.1	94.4	95.8	95.8	95.8	95.8	97.2	97.2	97.2	97.2	97.2	98.6
$M_{0.5}$	94.4	94.4	95.8	95.8	95.8	95.8	97.2	97.2	97.2	97.2	97.2	98.6
$M_{0.8}$	90.3	95.8	94.4	95.8	97.2	97.2	98.6	98.6	98.6	98.6	98.6	98.6
$M_{1.0}$	88.9	94.4	98.6	97.2	97.2	97.2	97.2	98.6	97.2	97.2	98.6	98.6
$\chi^{1.5}$	93.1	93.1	95.8	95.8	97.2	98.6	95.8	98.6	98.6	97.2	97.2	97.2
$\chi^{2.0}$	93.1	93.1	95.8	95.8	97.2	98.6	97.2	97.2	97.2	97.2	97.2	97.2
$\chi^{3.0}$	93.1	95.8	95.8	95.8	97.2	97.2	97.2	95.8	97.2	97.2	97.2	97.2
$\chi^{4.0}$	93.1	94.4	95.8	94.4	94.4	95.8	95.8	97.2	97.2	97.2	97.2	97.2
$R_{0.2}$	91.7	94.4	95.8	94.4	97.2	97.2	97.2	97.2	97.2	97.2	97.2	98.6
$R_{0.5}$	94.4	94.4	95.8	95.8	95.8	95.8	97.2	97.2	97.2	97.2	97.2	98.6
$R_{0.8}$	94.4	94.4	94.4	95.8	95.8	95.8	97.2	97.2	95.8	97.2	98.6	98.6
$R_{1.0}$	94.4	94.4	94.4	94.4	97.2	97.2	97.2	95.8	97.2	94.4	94.4	97.2
$R_{1.5}$	94.4	94.4	94.4	95.8	98.6	97.2	97.2	97.2	97.2	95.8	97.2	97.2
$R_{2.0}$	94.4	93.1	94.4	93.1	97.2	97.2	97.2	97.2	97.2	97.2	97.2	97.2
$R_{3.0}$	94.4	94.4	94.4	97.2	94.4	97.2	97.2	97.2	97.2	97.2	97.2	97.2
$R_{4.0}$	94.4	91.7	94.4	95.8	97.2	94.4	95.8	95.8	95.8	97.2	97.2	98.6
U	94.4	94.4	94.4	94.4	95.8	95.8	97.2	98.6	98.6	94.4	97.2	97.2

$R_{0.2}$-, and $M_{0.2}$-information measures, the index $\mathscr{S}$ varies similarly, while mutual information and normalized mutual information provide similar values of this index. The results also establish the fact that as the number of selected genes increases, the class separability index $\mathscr{S}$ decreases in case of I_α-, R_α-, and M_α-information measures for $\alpha = 0.2$, and ultimately it saturates. However, for a fixed number of selected genes, the values of $\mathscr{S}$ index in case of these three measures are lower than that of mutual information and normalized mutual information.

Finally, Table 5.10 reports the comparative performance of several information theoretic measures with respect to the class separability index $\mathscr{S}$. From all the results reported in Table 5.10, it can be seen that the class separability index $\mathscr{S}$ obtained using $I_{0.2}$-, $M_{0.2}$-, and $R_{0.2}$-information measures are better than those obtained using $I_{1.0}$ (mutual information) and U (normalized mutual information) for breast cancer, leukemia, and colon cancer data sets.

Table 5.7 Performance on colon cancer data set using NB classifier

f-Information measures	Number of selected genes											
	2	5	8	10	15	20	25	30	35	40	45	50
$I_{0.2}$	72.6	85.5	90.3	91.9	96.8	98.4	100	100	100	100	100	100
$I_{0.5}$	83.9	93.6	93.6	91.9	95.2	93.6	93.6	96.8	96.8	96.8	95.2	95.2
$I_{0.8}$	83.9	90.3	93.6	91.9	93.6	93.6	93.6	93.6	91.9	93.6	93.6	93.6
$I_{1.0}$	83.9	90.3	90.3	91.9	93.6	93.6	93.6	91.9	93.6	93.6	91.9	91.9
$I_{1.5}$	83.9	90.3	91.9	90.3	93.6	93.6	93.6	93.6	91.9	91.9	90.3	88.7
$I_{2.0}$	83.9	93.6	90.3	90.3	93.6	93.6	93.6	91.9	91.9	90.3	88.7	88.7
$I_{3.0}$	83.9	93.6	91.9	91.9	93.6	91.9	91.9	91.9	90.3	88.7	88.7	88.7
$I_{4.0}$	83.9	93.6	91.9	93.6	91.9	91.9	93.6	91.9	91.9	91.9	88.7	87.1
$M_{0.2}$	72.6	80.7	91.9	91.9	95.2	95.2	98.4	100	100	100	100	100
$M_{0.5}$	83.9	93.6	93.6	91.9	95.2	93.6	93.6	96.8	96.8	96.8	95.2	95.2
$M_{0.8}$	85.5	85.5	90.3	88.7	88.7	90.3	91.9	93.6	90.3	90.3	90.3	90.3
$M_{1.0}$	85.5	85.5	90.3	93.6	87.1	90.3	88.7	90.3	88.7	87.1	88.7	90.3
$\chi^{1.5}$	77.4	88.7	91.9	91.9	91.9	90.3	95.2	91.9	90.3	91.9	90.3	88.7
$\chi^{2.0}$	83.9	93.6	90.3	90.3	93.6	93.6	93.6	91.9	91.9	90.3	88.7	88.7
$\chi^{3.0}$	83.9	93.6	91.9	91.9	93.6	93.6	95.2	93.6	95.2	91.9	88.7	87.1
$\chi^{4.0}$	83.9	93.6	95.2	93.6	95.2	95.2	95.2	93.6	91.9	93.6	91.9	90.3
$R_{0.2}$	72.6	85.5	90.3	93.6	95.2	98.4	100	100	100	100	100	98.4
$R_{0.5}$	83.9	93.6	93.6	91.9	95.2	93.6	95.2	96.8	96.8	96.8	95.2	95.2
$R_{0.8}$	83.9	90.3	93.6	91.9	93.6	93.6	93.6	93.6	91.9	93.6	93.6	93.6
$R_{1.0}$	83.9	90.3	90.3	91.9	93.6	93.6	93.6	91.9	93.6	93.6	91.9	91.9
$R_{1.5}$	83.9	90.3	91.9	90.3	93.6	93.6	93.6	93.6	91.9	93.6	90.3	90.3
$R_{2.0}$	83.9	93.6	90.3	91.9	91.9	93.6	93.6	93.6	91.9	91.9	90.3	88.7
$R_{3.0}$	83.9	93.6	91.9	91.9	90.3	95.2	93.6	91.9	91.9	91.9	88.7	88.7
$R_{4.0}$	83.9	93.6	88.7	91.9	90.3	90.3	90.3	91.9	91.9	91.9	90.3	88.7
U	83.9	90.3	87.1	90.3	93.6	93.6	93.6	95.2	93.6	93.6	93.6	93.6

Table 5.8 Performance on colon cancer data set using SVM

f-Information measures	Number of selected genes											
	2	5	8	10	15	20	25	30	35	40	45	50
$I_{0.2}$	83.9	80.7	80.7	82.3	80.7	80.7	80.7	80.7	77.4	77.4	80.7	77.4
$I_{0.5}$	83.9	79.0	87.1	85.5	87.1	80.7	82.3	80.7	79.0	79.0	79.0	79.0
$I_{0.8}$	83.9	83.9	85.5	83.9	80.7	80.7	80.7	79.0	74.2	77.4	79.0	80.6
$I_{1.0}$	83.9	83.9	83.9	88.7	80.7	82.3	75.8	79.0	80.7	79.0	83.9	83.9
$I_{1.5}$	83.9	80.7	87.1	87.1	88.7	88.7	83.9	79.0	77.4	79.0	80.7	79.0
$I_{2.0}$	83.9	87.1	87.1	87.1	90.3	87.1	82.3	80.7	75.8	77.4	80.7	80.7
$I_{3.0}$	83.9	87.1	88.7	85.5	88.7	80.7	80.7	75.8	75.8	79.0	80.7	77.4
$I_{4.0}$	83.9	87.1	88.7	87.1	85.5	75.8	85.5	80.7	77.4	79.0	75.8	77.4
$M_{0.2}$	83.9	75.8	77.4	87.1	82.3	80.7	80.7	77.4	79.0	75.8	71.0	71.0
$M_{0.5}$	83.9	79.0	87.1	85.5	87.1	80.7	82.3	80.7	79.0	79.0	79.0	79.0
$M_{0.8}$	79.0	83.9	87.1	82.3	82.3	88.7	79.0	75.8	72.6	80.7	75.8	80.7
$M_{1.0}$	79.0	83.9	87.1	85.5	91.9	83.9	82.3	75.8	71.0	69.4	77.4	77.4
$\chi^{1.5}$	77.4	91.9	85.5	85.5	90.3	83.9	82.3	79.0	75.8	77.4	80.7	80.7

(continued)

Table 5.8 (continued)

f-Information measures	Number of selected genes											
	2	5	8	10	15	20	25	30	35	40	45	50
$\chi^{2.0}$	83.9	87.1	87.1	87.1	90.3	87.1	82.3	80.7	75.8	77.4	80.7	80.7
$\chi^{3.0}$	83.9	87.1	85.5	88.7	85.5	82.3	87.1	75.8	87.1	79.0	77.4	77.4
$\chi^{4.0}$	83.9	87.1	83.9	80.7	83.9	80.7	87.1	77.4	79.0	75.8	74.2	79.0
$R_{0.2}$	83.9	83.9	74.2	79.0	82.3	80.7	80.7	75.8	80.7	77.4	80.7	75.8
$R_{0.5}$	83.9	79.0	87.1	85.5	87.1	80.7	83.9	80.7	79.0	79.0	79.0	75.8
$R_{0.8}$	83.9	83.9	85.5	83.9	80.7	80.7	80.7	79.0	74.2	77.4	79.0	82.3
$R_{1.0}$	83.9	83.9	83.9	88.7	80.7	82.3	75.8	79.0	80.7	79.0	83.9	83.9
$R_{1.5}$	83.9	80.7	87.1	87.1	88.7	88.7	83.9	79.0	75.8	77.4	80.7	80.7
$R_{2.0}$	83.9	87.1	87.1	88.7	91.9	87.1	80.7	77.4	77.4	79.0	80.7	80.7
$R_{3.0}$	83.9	87.1	88.7	88.7	85.5	83.9	83.9	72.6	75.8	77.4	77.4	82.3
$R_{4.0}$	83.9	87.1	87.1	85.5	80.7	77.4	83.9	74.2	75.8	72.6	83.9	79.0
U	83.9	83.9	77.4	85.5	83.9	82.3	82.3	85.5	80.7	75.8	82.3	82.3

Table 5.9 Performance on colon cancer data set using K-NN rule

f-Information measures	Number of selected genes											
	2	5	8	10	15	20	25	30	35	40	45	50
$I_{0.2}$	82.3	82.3	77.4	74.2	79.0	82.3	75.8	75.8	82.3	85.5	83.9	83.9
$I_{0.5}$	83.9	77.4	75.8	79.0	83.9	87.1	88.7	85.5	85.5	85.5	87.1	83.9
$I_{0.8}$	83.9	74.2	85.5	85.5	85.5	85.5	87.1	87.1	87.1	88.7	88.7	85.5
$I_{1.0}$	83.9	74.2	82.3	88.7	87.1	87.1	87.1	87.1	87.1	87.1	87.1	87.1
$I_{1.5}$	83.9	87.1	85.5	85.5	88.7	88.7	88.7	88.7	87.1	87.1	87.1	87.1
$I_{2.0}$	83.9	85.5	85.5	85.5	88.7	88.7	88.7	88.7	87.1	88.7	88.7	87.1
$I_{3.0}$	83.9	85.5	87.1	85.5	87.1	88.7	87.1	87.1	88.7	88.7	88.7	88.7
$I_{4.0}$	83.9	85.5	87.1	88.7	90.3	87.1	88.7	88.7	88.7	87.1	88.7	87.1
$M_{0.2}$	82.3	80.7	79.0	77.4	79.0	80.7	74.2	75.8	75.8	74.2	79.0	77.4
$M_{0.5}$	83.9	77.4	75.8	79.0	83.9	87.1	88.7	85.5	85.5	85.5	87.1	83.9
$M_{0.8}$	83.9	83.9	85.5	85.5	83.9	88.7	82.3	87.1	83.9	85.5	87.1	87.1
$M_{1.0}$	83.9	83.9	88.7	87.1	87.1	87.1	85.5	83.9	83.9	83.9	85.5	88.7
$\chi^{1.5}$	79.0	88.7	88.7	85.5	87.1	85.5	83.9	85.5	85.5	88.7	88.7	88.7
$\chi^{2.0}$	83.9	85.5	85.5	85.5	88.7	88.7	88.7	88.7	87.1	88.7	88.7	87.1
$\chi^{3.0}$	83.9	85.5	85.5	88.7	90.3	88.7	88.7	88.7	87.1	90.3	85.5	85.5
$\chi^{4.0}$	83.9	85.5	88.7	88.7	87.1	87.1	87.1	85.5	85.5	85.5	85.5	85.5
$R_{0.2}$	82.3	75.8	82.3	82.3	74.2	79.0	75.8	75.8	75.8	85.5	83.9	82.3
$R_{0.5}$	83.9	77.4	75.8	79.0	83.9	83.9	88.7	87.1	85.5	85.5	87.1	83.9
$R_{0.8}$	83.9	74.2	85.5	85.5	85.5	85.5	87.1	87.1	87.1	88.7	88.7	88.7
$R_{1.0}$	83.9	74.2	82.3	88.7	87.1	87.1	87.1	87.1	87.1	87.1	87.1	87.1
$R_{1.5}$	83.9	87.1	85.5	85.5	88.7	88.7	88.7	88.7	87.1	87.1	88.7	87.1
$R_{2.0}$	83.9	85.5	85.5	88.7	87.1	88.7	87.1	87.1	87.1	88.7	88.7	87.1
$R_{3.0}$	83.9	85.5	87.1	88.7	87.1	88.7	87.1	87.1	87.1	88.7	88.7	88.7
$R_{4.0}$	83.9	85.5	85.5	85.5	87.1	85.5	88.7	87.1	87.1	88.7	88.7	87.1
U	83.9	74.2	82.3	83.9	85.5	87.1	85.5	87.1	85.5	88.7	85.5	87.1

Table 5.10 Performance on three cancer data sets using class separability index

Data Set	$I_{1.0}/R_{1.0}$	U	$I_{0.2}$	$R_{0.2}$	$M_{0.2}$
Breast cancer	3.302	3.314	3.285	3.308	3.297
Leukemia	2.000	2.194	1.906	1.876	1.871
Colon cancer	0.718	0.716	0.463	0.449	0.547

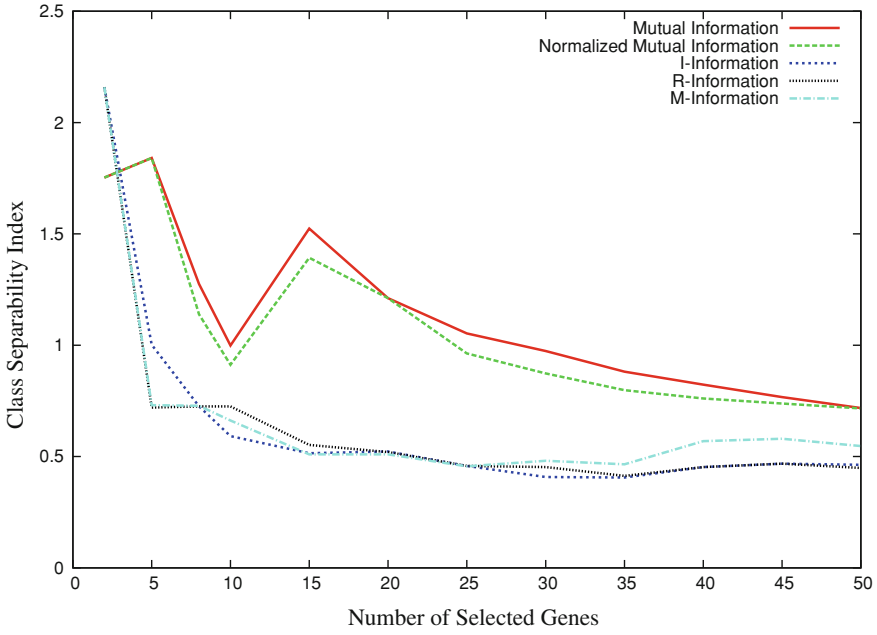

Fig. 5.1 Variation of class separability index with respect to number of selected genes for colon cancer data set

5.4 Conclusion and Discussion

This chapter introduces different f-information measures in order to identify discriminative genes from high dimensional gene expression data. It presents the results of selecting relevant and nonredundant genes from microarray data using different measures from information theory. The popular and extensively researched measure of mutual information is compared with V-, I_α-, χ^α-, and R_α-information measures. All information theoretic measures denote the divergence of the joint distribution of the genes' expression values from the joint distribution for complete independence of the genes.

The minimum redundancy-maximum relevance framework is used as the gene selection method for different f-information measures. The performance of different measures is evaluated by the predictive accuracy of naive Bayes classifier, K-nearest neighbor rule, and support vector machine. For all data sets, significantly better results are found for several measures, compared to mutual information.

The results obtained on real data sets demonstrate that the reported f-information measures can bring a remarkable improvement on gene selection problem, and therefore, the f-information measures can be a promising alternative to mutual information for gene selection. They are capable of identifying discriminative genes that may contribute to revealing underlying class structures, providing a useful tool for the exploratory analysis of biological data.

The application of different information measures other than mutual information shows promise in achieving better gene selection results. Although some measures are more difficult to optimize, I_α-, R_α-, and M_α-information measures are shown to produce more accurate results for the selection of relevant and nonredundant genes from microarray gene expression data. The smoothness of the gene selection function, that is, the function to be optimized in order to find the relevant and nonredundant genes, is influenced by the value of α. However, I_α and R_α equal mutual information for the limit $\alpha \to 1$, it may be beneficial to start selection with $\alpha = 1$ to take advantage of the smoothness of the function and to adapt the value of α in subsequent iterations for better accuracy. The optimal value of α may differ per microarray data set.

The various f-information measures are only used to one representative gene selection method, that is, minimum redundancy-maximum relevance framework. In future, these measures can be extended to other gene selection methods and further their merits and limitations may be evaluated. In order to address the problem of multiplicity of marker genes, a detailed analysis of the biological relevance of the selected genes can be conducted. The gene interactions can be studied in detail to see whether incorporation of the gene interaction information can improve the diagnostic test.

Both Chaps. 4 and 5 present feature selection algorithms based on maximum relevance-maximum significance (MRMS) and minimum redundancy-maximum relevance (mRMR) criteria, respectively. However, the theory of rough sets is used in Chap. 4 to compute the MRMS criterion, while mutual information and several other f-information measures are used to calculate the mRMR criterion in the current chapter. Next two chapters report the comparative performance analysis of both MRMS and mRMR criteria for feature selection. In Chap. 6, mutual information is used for computing both mRMR and MRMS criteria. On the other hand, the MRMS criterion is calculated using rough set theory in Chap. 7, while mutual information is used for the mRMR criterion.

References

1. Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc Nat Acad Sci USA 96(12):6745–6750
2. Baldi P, Long AD (2001) A bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes. Bioinformatics 17(6):509–519
3. Ben-Dor A, Bruhn L, Friedman N, Nachman I, Schummer M, Yakhini Z (2000) Tissue classification with gene expression profiles. J Comput Biol 7(3/4):559–584

4. Blanco R, Larranaga P, Inza I, Sierra B (2004) Gene selection for cancer classification using wrapper approaches. Int J Pattern Recognit Artif Intell 18(8):1373–1390
5. Bø T, Jonassen I (2002) New feature subset selection procedures for classification of expression profiles. Genome Biol 3(4):17
6. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
7. Das SK (1971) Feature selection with a linear dependence measure. IEEE Trans Comput 20(9):1106–1109
8. Dash M, Liu H (2000) Unsupervised feature selection. In: Proceedings of Pacific Asia conference on knowledge discovery and data mining, pp 110–121
9. Devijver PA, Kittler J (1982) Pattern recognition: a statistical approach. Prentice Hall, Englewood Cliffs
10. Ding C, Peng H (2003) Minimum redundancy feature selection from microarray gene expression data. In: Proceedings of the international conference on computational systems, Bioinformatics, pp 523–528
11. Domany E (2003) Cluster analysis of gene expression data. J Stat Phys 110(3–6):1117–1139
12. Duda RO, Hart PE, Stork DG (1999) Pattern classification and scene analysis. Wiley, New York
13. Dudoit S, Fridlyand J, Speed TP (2002) Comparison of discrimination methods for the classification of tumors using gene expression data. J Am Stat Assoc 97(457):77–87
14. Efron B, Tibshirani R, Storey JD, Tusher V (2001) Empirical Bayes analysis of a microarray experiment. J Am Stat Assoc 96(456):1151–1160
15. Fox R, Dimmic M (2006) A two-sample Bayesian *t*-test for microarray data. BMC Bioinformatics 7(1):126
16. Gevaert O, Smet FD, Timmerman D, Moreau Y, Moor BD (2006) Predicting the prognosis of breast cancer by integrating clinical and microarray data with Bayesian networks. Bioinformatics 22(14):e184–e190
17. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, Lander ES (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286(5439):531–537
18. Gruzdz A, Ihnatowicz A, Slezak D (2006) Interactive gene clustering—a case study of breast cancer microarray data. Inf Syst Front 8:21–27
19. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46(1–3):389–422
20. Hall MA (2000) Correlation-based feature selection for discrete and numeric class machine learning. In: Proceedings of the 17th international conference on machine learning, pp 359–366
21. Heydorn RP (1971) Redundancy in feature extraction. IEEE Trans Comput 20(9):1051–1054
22. Heyer LJ, Kruglyak S, Yooseph S (1999) Exploring expression data: identification and analysis of coexpressed genes. Genome Res 9(11):1106–1115
23. Hu Q, Pan W, An S, Ma P, Wei J (2010) An efficient gene selection technique for cancer recognition based on neighborhood mutual information. Int J Mach Learn Cybern 1(1–4):63–74
24. Inza I, Larranaga P, Blanco R, Cerrolaza AJ (2004) Filter versus wrapper gene selection approaches in DNA microarray domains. Artif Intell Med 31(2):91–103
25. Jafari P, Azuaje F (2006) An assessment of recently published gene expression data analyses: reporting experimental design and statistical factors. BMC Med Inform Decis Mak 6(1):27
26. Jain AK, Dubes RC (1988) Algorithms clustering data. Prentice Hall, Englewood Cliffs
27. Jiang D, Tang C, Zhang A (2004) Cluster analysis for gene expression data: a survey. IEEE Trans Knowl Data Eng 16(11):1370–1386
28. Jiang H, Deng Y, Chen HS, Tao L, Sha Q, Chen J, Tsai CJ, Zhang S (2004) Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. BMC Bioinformatics 5(1):81
29. Jirapech-Umpai T, Aitken S (2005) Feature selection and classification for microarray data analysis: evolutionary methods for identifying predictive genes. BMC Bioinformatics 6(1):148

30. Kiranagi BB, Guru DS, Ichino M (2007) Exploitation of multivalued type proximity for symbolic feature selection. In: Proceedings of the international conference on computing: theory and applications, 2007
31. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97(1–2):273–324
32. Kononenko I, Simec E, Sikonja MR (1997) Overcoming the myopia of inductive learning algorithms with RELIEFF. Appl Intell 7:39–55
33. Lee JW, Lee JB, Park M, Song SH (2005) An extensive comparison of recent classification tools applied to microarray data. Comput Stat Data Anal 48(4):869–885
34. Li J, Su H, Chen H, Futscher BW (2007) Optimal search-based gene subset selection for gene array cancer classification. IEEE Trans Inf Technol Biomed 11(4):398–405
35. Li L, Weinberg CR, Darden TA, Pedersen LG (2001) Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. Bioinformatics 17(12):1131–1142
36. Li T, Zhang C, Ogihara M (2004) A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. Bioinformatics 20(15):2429–2437
37. Liao JG, Chin KV (2007) Logistic regression for disease classification using microarray data: model selection in a large p and small n case. Bioinformatics 23(15):1945–1951
38. Liu Q, Sung A, Chen Z, Liu J, Chen L, Qiao M, Wang Z, Huang X, Deng Y (2011) Gene selection and classification for cancer microarray data based on machine learning and similarity measures. BMC Genomics 12(Suppl 5):S1
39. Liu X, Krishnan A, Mondry A (2005) An entropy based gene selection method for cancer classification using microarray data. BMC Bioinformatics 6(1):76
40. Loennstedt I, Speed TP (2002) Replicated microarray data. Statistica Sinica 12:31–46
41. Lyons-Weiler J, Patel S, Becich M, Godfrey T (2004) Tests for finding complex patterns of differential expression in cancers: towards individualized medicine. BMC Bioinformatics 5(1):110
42. Ma S, Huang J (2005) Regularized ROC method for disease classification and biomarker selection with microarray data. Bioinformatics 21(24):4356–4362
43. Maji P (2009) f-information measures for efficient selection of discriminative genes from microarray data. IEEE Trans Biomed Eng 56(4):1063–1069
44. Maji P, Pal SK (2010) Fuzzy-rough sets for information measures and selection of relevant genes from microarray data. IEEE Trans Syst Man Cybern B Cybern 40(3):741–752
45. Mamitsuka H (2006) Selecting features in microarray classification using ROC curves. Pattern Recognit 39(12):2393–2404
46. McLachlan GJ, Do KA, Ambroise C (2004) Analyzing microarray gene expression data. Wiley, Hoboken
47. Mitra P, Murthy CA, Pal SK (2002) Unsupervised feature selection using feature similarity. IEEE Trans Pattern Anal Mach Intell 24(3):301–312
48. Miyano S, Imoto S, Sharma A (2012) A top-r feature selection algorithm for microarray gene expression data. IEEE/ACM Trans Comput Biol Bioinf 9(3):754–764
49. Newton MA, Kendziorski CM, Richmond CS, Blattner FR, Tsui KW (2001) On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data. J Comput Biol 8(1):37–52
50. Ooi CH, Tan P (2003) Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. Bioinformatics 19(1):37–44
51. Pan W (2003) On the use of permutation in and the performance of a class of nonparametric methods to detect differential gene expression. Bioinformatics 19(11):1333–1340
52. Pang H, George SL, Hui K, Tong T (2012) Gene selection using iterative feature elimination random forests for survival outcomes. IEEE/ACM Trans Comput Biol Bioinf 9(5):1422–1431
53. Park PJ, Pagano M, Bonetti M (2001) A nonparametric scoring algorithm for identifying informative genes from microarray data. In: Proceedings of Pacific symposium on biocomputing, pp. 52–63

54. Pavlidis P, Poirazi P (2006) Individualized markers optimize class prediction of microarray data. BMC Bioinformatics 7(1):345
55. Peng H, Long F, Ding C (2005) Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 27(8):1226–1238
56. Pluim JPW, Maintz JBA, Viergever MA (2004) f-information measures in medical image registration. IEEE Trans Med Imaging 23(12):1508–1516
57. Ruiz R, Riquelme JC, Ruiz JSA (2006) Incremental wrapper-based gene selection from microarray data for cancer classification. Pattern Recognit 39(12):2383–2392
58. Saeys Y, Inza I, Larraaga P (2007) A review of feature selection techniques in bioinformatics. Bioinformatics 23(19):2507–2517
59. Shah M, Marchand M, Corbeil J (2012) Feature selection with conjunctions of decision stumps and learning from microarray data. IEEE Trans Pattern Anal Mach Intell 34(1):174–186
60. Sharma A, Imoto S, Miyano S, Sharma V (2012) Null space based feature selection method for gene expression data. Int J Mach Learn Cybern 3(4):269–276
61. Slavkov I, Gjorgjioski V, Struyf J, Deroski S (2010) Finding explained groups of time-course gene expression profiles with predictive clustering trees. Mol BioSyst 6:729–740
62. Statnikov A, Aliferis CF, Tsamardinos I, Hardin D, Levy S (2005) A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. Bioinformatics 21(5):631–643
63. Thomas JG, Olson JM, Tapscott SJ, Zhao LP (2001) An efficient and robust statistical modeling approach to discover differentially expressed genes using genomic expression profiles. Genome Res 11(7):1227–1236
64. Tusher V, Tibshirani R, Chu G (2001) Significance analysis of microarrays applied to the ionizing radiation response. Proc Nat Acad Sci USA 98:5116–5121
65. Uriarte RD, de Andres SA (2006) Gene selection and classification of microarray data using random forest. BMC Bioinformatics 7(1):3
66. Vajda I (1989) Theory of statistical inference and information. Kluwer Academic, Dordrecht
67. Vapnik V (1995) The nature of statistical learning theory. Springer, New York
68. Wang Y, Tetko IV, Hall MA, Frank E, Facius A, Mayer KFX, Mewes HW (2005) Gene selection from microarray data for cancer classification—a machine learning approach. Comput Biol Chem 29(1):37–46
69. West M, Blanchette C, Dressman H, Huang E, Ishida S, Spang R, Zuzan H, Olson JA, Marks JR, Nevins JR (2001) Predicting the clinical status of human breast cancer by using gene expression profiles. Proc Nat Acad Sci USA 98(20):11462–11467
70. Xing EP, Jordan MI, Karp RM (2001) Feature selection for high-dimensional genomic microarray data. In: Proceedings of the 18th international conference on machine learning, pp 601–608
71. Xiong M, Fang X, Zhao J (2001) Biomarker identification by feature wrappers. Genome Res 11(11):1878–1887
72. Yang F, Mao KZ (2011) Robust feature selection for microarray data based on multicriterion fusion. IEEE/ACM Trans Comput Biol Bioinf 8(4):1080–1092
73. Yeoh EJ, Ross ME, Shurtleff SA, Williams WK, Patel D, Mahfouz R, Behm FG, Raimondi SC, Relling MV, Patel A, Cheng C, Campana D, Wilkins D, Zhou X, Li J, Liu H, Pui CH, Evans WE, Naeve C, Wong L, Downing JR (2002) Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. Cancer Cell 1(2):133–143
74. Yeung K, Bumgarner R (2003) Multiclass classification of microarray data with repeated measurements: application to cancer. Genome Biol 4(12):R83

Chapter 6
Identification of Disease Genes Using Gene Expression and Protein–Protein Interaction Data

6.1 Introduction

Genetic diseases are caused by abnormalities in genes or chromosomes. Most genetic disorders are quite rare. A genetic disease may be heritable disorder or may not be. While some genetic diseases are passed down from the parent's genes, others are frequently caused by new mutations or changes to the DNA. In other instances, the same disease, for example, some forms of cancer, may stem from an inherited genetic condition in some people, from new mutations in some people, and from nongenetic causes in other people. As their name suggests, these diseases are caused by the dysfunction of some genes. Therefore, these genes are better known as disease genes [2]. Examples of diseases caused by dysfunction of a gene include Alzheimer's disease, breast cancer, leukemia, colorectal cancer, down syndrome, heart disease, and so forth.

The colorectal cancer, commonly known as colon cancer or bowel cancer, is a cancer from uncontrolled cell growth in the colon or rectum, a part of the large intestine. Globally, grater than 1 million people get affected by colorectal cancer yearly, resulting in about 0.5 million deaths. The colorectal cancer is the second most common cause of cancer in women and the third most common in men with it being the forth most common cause of cancer death. Therefore, early detection of colorectal cancer can help in increasing the patient survival chance and may also help to improve the prognosis. Hence, it is important to identify colon cancer-related genes in order to improve and develop the diagnostic tools.

Recent advancement and wide use of high-throughput biotechnologies have been producing huge amount of data such as yeast two-hybrid system, protein complex, gene expression profile, and so forth. These data sets have been widely used in different studies to understand the function of disease genes [6, 8, 21–24]. Analyzing the difference of gene expression levels in particular cell types may provide an idea about the propensity of a disease. Specifically, if a set of genes shows a consistent pattern of different expression levels in sick subjects and a control group, then that gene set is likely a strong candidate of playing a pathogenic role.

P. Maji and S. Paul, *Scalable Pattern Recognition Algorithms*,
DOI: 10.1007/978-3-319-05630-2_6,

Differences in expression levels can be detected primarily by microarray studies [14, 36, 37, 42, 46]. In this background, microarray gene expression data has been widely used for identification of disease genes. Different feature selection algorithms, discussed in Chaps. 4 and 5, can be used to identify disease genes from microarray gene expression data.

In [4, 19, 52], it has been shown that the genes associated with the same disorder tend to share common functional features, reflecting that their protein products have a tendency to interact with each other. Hence, another indicative trait of a disease gene is that its protein product is strongly linked to other disease-gene proteins. In this background, the protein–protein interaction (PPI) data have been used in various studies to identify disease genes [7, 30, 39, 43]. Individually microarray data or the PPI network data can be used to identify potential disease genes, although there is a limited chance of finding novel disease genes from such an analysis. In this regard, data integration methods have been developed to identify pleiotropic genes involved in the physiological cellular processes of many diseases.

The integrated approaches assume that the protein products of disease genes tend to be in close to differentially expressed genes in the protein interaction network. This type of problem has been observed as equivalent to the set cover problem in graph theory, which is NP-complete [28]. Hence, such a large-scale protein networks can only be analyzed with approximate, greedy algorithms. Nitsch et al. [41] developed the concept of soft neighborhood, where each gene is given a contributing weight, which decreases with the distance from the candidate gene on the protein network. Wu et al. [51] developed a method by integrating gene expression data and the PPI network data to prioritize cancer-associated genes. Zhao et al. [53] also proposed an approach by integrating gene expression data and the PPI network data to select disease genes. Jia et al. [26] developed a dense module searching method to identify disease genes for complex diseases by integrating the association signal from genome wide association studies data sets into the human PPI network. Li and Li [34] developed another approach to identify candidate disease genes, where heterogeneous genomic and phenotype data sets are used. In this method, separate gene networks are first developed using different types of data sets. The various genomic networks are then merged into a single graph, and disease genes are identified using random walk. In [33], minimum redundancy-maximum relevance (mRMR) [16, 17] approach has been used to select a set of genes from expression data. The selected gene set is then used for identification of intermediate genes between a pair of selected genes using the PPI network data. However, the mRMR method selects a set of genes by maximizing the relevance and minimizing the redundancy among the selected genes. As the redundancy measure does not take into account the supervised information of class labels, the mRMR method may not be always effective for identification of disease genes.

In this regard, this chapter describes an insilico approach to identify disease genes associated with colorectal cancer. It uses both the gene expression and PPI data. A set of differentially expressed genes is first identified using a new gene selection algorithm, termed as MIMRMS, from microarray gene expression data. The MIMRMS algorithm judiciously integrates the merits of maximum relevance-

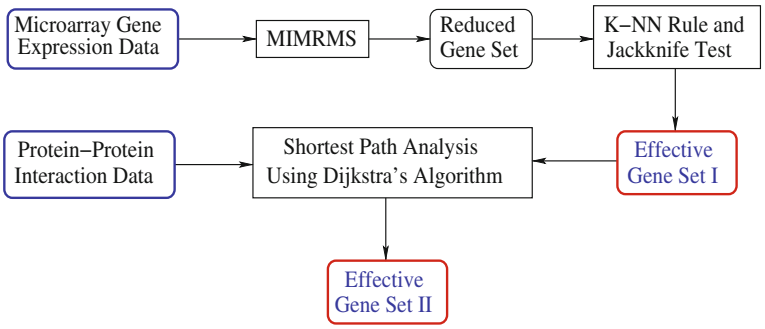

Fig. 6.1 Schematic flow diagram of the insilico approach for identification of disease genes

maximum significance (MRMS) criterion, mentioned in Chap. 4, and mutual information (MI). It selects a set of differentially expressed genes from microarray gene expression data by maximizing the relevance and significance of genes. Both relevance and significance are calculated using mutual information. The selected genes are then used to construct a protein association network using the PPI data. The Dijkstra's algorithm [15] is used to construct the shortest paths between a pair of genes selected by the MIMRMS method. Finally, a set of genes is identified as disease genes. The statistical analysis of the gene set, obtained from both gene expression and PPI data, establishes the fact that the identified genes are significantly linked with colorectal cancer.

The rest of the chapter is organized as follows: Sect. 6.2 presents the detailed description of the integrated approach. Experimental results, a brief description of expression data and protein–protein interaction data, and comparison among different algorithms are mentioned in Sect. 6.3. The statistical significance analysis of the identified disease genes with respect to the known disease genes are also reported in this section. Concluding remarks are given in Sect. 6.4.

6.2 Integrated Method for Identifying Disease Genes

In order to understand the etiology of a disease, identification of disease genes is a vital task. In this regard, this chapter presents a method, integrating judiciously both gene expression and PPI data. The integrated method has three main operational steps as illustrated in Fig. 6.1.

Selection of Differentially Expressed Genes

The first step of the integrated method selects a set $\mathbb{S}$ of differentially expressed genes from the whole gene set $\mathbb{C}$ of the given microarray gene expression data set.

The gene set $\mathbb{S}$ is selected using the MIMRMS method by maximizing both relevance and significance of genes present in $\mathbb{S}$. In general, the microarray data may contain a number of irrelevant and insignificant genes. The presence of such genes may lead to a reduction in the useful information. On the other hand, a gene set with high relevance and high significance enhances the predictive capability. The current method uses maximum relevance-maximum significance criterion, reported in Chap. 4, to select the relevant and significant genes from high dimensional microarray gene expression data sets.

Let $\mathbb{C} = \{\mathscr{A}_1, \ldots, \mathscr{A}_i, \ldots, \mathscr{A}_j, \ldots, \mathscr{A}_m\}$ be the set of m genes of a given microarray gene expression data set and $\mathbb{S}$ is the set of selected genes. Define $\gamma_{\mathscr{A}_i}(\mathbb{D})$ as the relevance of the gene $\mathscr{A}_i$ with respect to the class labels $\mathbb{D}$ while $\sigma_{\{\mathscr{A}_i,\mathscr{A}_j\}}(\mathbb{D}, \mathscr{A}_j)$ as the significance of the gene $\mathscr{A}_j$ with respect to the set $\{\mathscr{A}_i, \mathscr{A}_j\}$. The total relevance of all selected genes is $\mathscr{J}_{\text{relev}} = \sum_{\mathscr{A}_i \in \mathbb{S}} \gamma_{\mathscr{A}_i}(\mathbb{D})$, while the total significance among the selected genes is $\mathscr{J}_{\text{signf}} = \sum_{\mathscr{A}_i \neq \mathscr{A}_j \in \mathbb{S}} \sigma_{\{\mathscr{A}_i,\mathscr{A}_j\}}(\mathbb{D}, \mathscr{A}_j)$. Hence, the problem of selecting a set $\mathbb{S}$ of relevant and significant genes from the whole set $\mathbb{C}$ of m genes, as reported in Chap. 4, is equivalent to maximize both $\mathscr{J}_{\text{relev}}$ and $\mathscr{J}_{\text{signf}}$, that is, to maximize the objective function $\mathscr{J} = \mathscr{J}_{\text{relev}} + \beta \mathscr{J}_{\text{signf}}$, where β is a weight parameter. To solve the above problem, the greedy algorithm, reported in Chap. 4, is used in the current study. Both the relevance and significance of a gene are calculated based on the theory of mutual information [50], as described in Chap. 5, while the definition of significance is exactly same as (4.7) of Chap. 4 or (9.7) of Chap. 9.

Selection of Effective Gene Set I

In the second step, a set of effective genes are identified as disease genes. The effective gene set I, as mentioned in Fig. 6.1 and denoted by $\mathbb{S}_{\text{GE}}$, is a subset of $\mathbb{S}$, and defined as the gene set for which the prediction model or classifier attains its maximum classification accuracy. The K-nearest neighbor (K-NN) rule [18] is used here for evaluating the effectiveness of the reduced gene set for classification. A brief description of the K-NN rule is reported in Chap. 5. The value of K, chosen for the current study, is 1, while the dissimilarity between two samples is calculated as follows:

$$D(x_i, x_j) = 1 - \frac{x_i \cdot x_j}{||x_i|| \cdot ||x_j||} \tag{6.1}$$

where x_i and x_j are two vectors representing two tissue samples, $x_i \cdot x_j$ is their dot product, and $||x_i||$ and $||x_j||$ are their moduli. The smaller the $D(x_i, x_j)$, the more similar the two samples are.

To calculate the classification accuracy of the K-NN rule, the jackknife test [45] is used, although both independent data set test and subsampling test can also be used. However, jackknife estimators allow to correct for a bias and its statistical error. In the jackknife test, all the samples in the given data set are singled out one-by-one

and tested by the classifier trained by the remaining samples. During the process of jackknifing, both the training and testing data sets are actually open, and each sample is in turn moved between the two. The jackknife method is recommended as the standard for error bar calculation. In unbiased situation, the jackknife and the usual error bars agree. Otherwise, the jackknife estimates are improvements, so that one cannot loose. In particular, the jackknife method solves the question of error propagation elegantly and with little efforts involved. Also, it is very much applicable for the data sets with small number of training samples and large number of features or genes. Therefore, in this work, jackknife test is used to evaluate the prediction capability of the K-NN rule.

Selection of Effective Gene Set II

Finally, the effective gene set II, denoted by $\mathbb{S}_{\text{GE+PPI}}$, is obtained from the PPI data based on the set $\mathbb{S}_{\text{GE}}$, the effective gene set I. It has been observed that proteins with short distances to each other in the network are more likely to involve in the common biological functions [5, 31, 40, 48], and that interactive neighbors are more likely to have identical biological function than noninteractive ones [27, 32]. This is because the query protein and its interactive proteins may form a protein complex to perform a particular function or involved in a same pathway.

The Search Tool for the Retrieval of Interacting Genes (STRING) [49] is an online database resource that provides both experimental as well as predicted interaction information with a confidence score. In general, the graph is a very useful tool for studying complex biological systems as it can provide intuitive insights and the overall structure property, as demonstrated by various studies on a series of important biological topics [1, 3, 9–13, 54, 55]. In this work, after selecting the gene set $\mathbb{S}_{\text{GE}}$, a graph $G(V, E)$ is constructed with the PPI data from the STRING using the gene set $\mathbb{S}_{\text{GE}}$. In between each pair of genes, an edge is assigned in the graph. The weight of the edge E in graph G is derived from the confidence score according to the relation $\omega^G = 1000 \times (1 - \omega^0)$, where ω^G is the weight in graph G while ω^0 is the confidence score between two proteins concerned. Accordingly, a functional protein association network with edge weight is generated. In order to identify the shortest path from each of the selected differentially expressed genes of $\mathbb{S}_{\text{GE}}$ to remaining genes of the set $\mathbb{S}_{\text{GE}}$ in the graph, Dijkstra's algorithm [15] is used. Finally, the genes present in the shortest path are picked up and ranked according to their betweenness value. Let this set of genes be $\mathbb{S}_{\text{PPI}}$. The effective gene set II, that is, $\mathbb{S}_{\text{GE+PPI}}$, is the union of sets $\mathbb{S}_{\text{GE}}$ and $\mathbb{S}_{\text{PPI}}$, that is, $\mathbb{S}_{\text{GE+PPI}} = \mathbb{S}_{\text{GE}} \cup \mathbb{S}_{\text{PPI}}$.

6.3 Experimental Results

In the current integrated method, the disease genes are identified by using both gene expression and PPI data sets. The mutual information-based maximum relevance-maximum significance (MIMRMS) method is used to select differentially expressed

genes from microarray data. On the other hand, the method proposed by Li et al. [33] uses minimum redundancy-maximum relevance (mRMR) framework [16, 17]. However, one may also use maximum relevance (MR) method. This section presents the comparative performance analysis of the MIMRMS, mRMR, and MR algorithms. The effectiveness of different algorithms are shown using integrated data consisting of both colorectal gene expression and PPI data.

For colorectal cancer expression data set, 50 top-ranked genes are selected by each gene selection algorithm for further analysis. The jackknife test is used to compute the classification accuracy of the K-NN rule. Based on the accuracy, the effective gene set $\mathbb{S}_{\text{GE}}$ is identified for each gene selection algorithm. Next, the PPI network is constructed using the gene set $\mathbb{S}_{\text{GE}}$, and the effective gene set $\mathbb{S}_{\text{GE+PPI}}$ is obtained based on the shortest path analysis of the constructed PPI network. Finally, the statistical significance analysis is performed on each identified gene set with respect to both known cancer and colorectal cancer genes.

6.3.1 Gene Expression Data Set Used

In this study, the gene expression data from the colorectal cancer study of Hinoue et al. [20] is used. The gene expression profiling of 26 colorectal tumors and matches histologically normal adjacent colonic tissue samples were retrieved from the NCBI Gene Expression Omnibus (http://www.ncbi.nlm.nih.gov/geo/) with the accession number of GSE25070. The number of genes and samples in this data set are 24526 and 52, respectively. The data set is preprocessed by standardizing each sample to zero mean and unit variance.

6.3.2 Identification of Differentially Expressed Genes

Figure 6.2 represents the predictive accuracy of the K-NN rule obtained using the MR, mRMR, and MIMRMS algorithms. From the figure, it can be seen that the MR and mRMR methods attain 100 % classification accuracy with 8 and 6 genes, respectively, while the MIMRMS method achieves this accuracy with 20 genes. The statistical significance analysis report next confirms that both MR and mRMR methods overestimate the classification accuracy of the K-NN rule compared to the MIMRMS method. In effect, the MIMRMS method is able to find more significant effective gene set compared to both MR and mRMR methods.

6.3.3 Overlap with Known Disease-Related Genes

The gene set $\mathbb{S}_{\text{GE}}$ selected by the MIMRMS method is compared with the gene sets $\mathbb{S}_{\text{GE}}$ obtained by both the MR and mRMR methods, in terms of the degree of

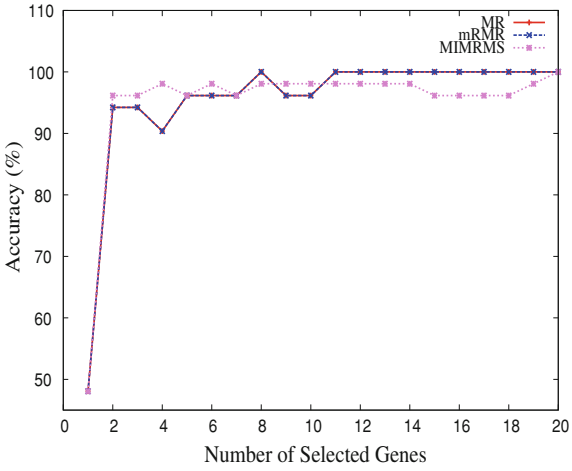

Fig. 6.2 Classification accuracy obtained using different gene selection algorithms

overlapping with three gene lists, namely, LIST-1, LIST-2, and LIST-3. The LIST-1 contains 742 cancer-related genes, which are collected from the Cancer Gene Census of the Sanger Centre, Atlas of Genetics and Cytogenetic in Oncology [25], and Human Protein Reference Database [29]. On the other hand, both LIST-2 and LIST-3 consist of colorectal cancer-related genes. While the LIST-2 is retrieved from the study of Sabatas-Bellver et al. [47], the LIST-3 is prepared from the work of Nagaraj and Reverter [38]. While LIST-2 contains 438 colorectal cancer genes, LIST-3 consists of 134 colorectal cancer genes.

The MR method attains highest predictive accuracy with eight genes. Hence, the selected gene set $\mathbb{S}_{GE}$ of the MR method contains eight genes, namely, **GUCA2B**, **BEST2**, **TMIGD**, **CLDN8**, **PI16**, **SCNN1B**, **CLCA4**, and **ADH1B**. Out of these eight genes, only **SCNN1B** overlaps with the LIST-1. On the other hand, five genes, namely, **GUCA2B**, **CLDN8**, **SCNN1B**, **CLCA4**, and **ADH1B**, overlap with the LIST-2, while only **GUCA2B** overlaps with the LIST-3. Similarly, the gene set $\mathbb{S}_{GE}$ of the mRMR method consists of six genes, namely, **CDH3**, **PI16**, **GUCA2B**, **HMGCLL1**, **BEST2**, and **SPIB**, as the mRMR method achieves highest predictive accuracy with these genes. However, none of them overlaps with the LIST-1. Out of six genes, three genes, namely, **CDH3**, **GUCA2B**, and **SPIB**, overlap with the LIST-2, while two genes, namely, **GUCA2B** and **SPIB**, overlap with the LIST-3.

On the other hand, the MIMRMS method provides 100 % classification accuracy of the K-NN rule with 20 genes. Hence, the gene set $\mathbb{S}_{GE}$ corresponding to the MIMRMS method consists of 20 genes, namely, **GUCA2B**, **PI16**, **CILP**, **SCNN1B**, **IL8**, **CA4**, **BCHE**, **BEST2**, **CLCA4**, **PECI**, **TMEM37**, **AFF3**, **CLDN8**, **ADH1B**, **CA1**, **GNG7**, **NR3C2**, **SCARA5**, **WISP2**, and **TMIGD**. Out of these 20 genes, three genes, namely, **CA4**, **AFF3**, and **NR3C2**, overlap with the LIST-1. On the other hand, eleven genes, namely, **GUCA2B**, **SCNN1B**, **IL8**, **CA4**, **BCHE**, **CLCA4**, **AFF3**,

Table 6.1 Overlap with LIST-1

Methods		Yes	No	Total
MR	Yes	1	7	8
	No	741	17742	18483
mRMR	Yes	0	6	6
	No	742	17743	18485
MIMRMS	Yes	3	17	20
	No	739	17732	18471
	Total	742	17749	18491

Table 6.2 Overlap with LIST-2

Methods		Yes	No	Total
MR	Yes	5	3	8
	No	433	20386	20819
mRMR	Yes	3	3	6
	No	435	20386	20821
MIMRMS	Yes	11	9	20
	No	427	20380	20807
	Total	438	20389	20827

CLDN8, **ADH1B**, **CA1**, and **SCARA5**, overlap with the LIST-2, while **GUCA2B**, **SCNN1B**, **IL8**, and **BCHE** overlap with the genes of the LIST-3.

Tables 6.1, 6.2, and 6.3 represent the statistical significance test of the gene sets $\mathbb{S}_{GE}$ selected by the MR, mRMR, and MIMRMS methods with respect to the genes of LIST-1, LIST-2, and LIST-3, respectively. In Table 6.1, the LIST-1 contains 742 cancer-related genes and the total number of genes in Illumina Ref-8 whole-genome expression Bead-Chip is 18491. Using the Fisher's exact test, statistical analysis of the overlapped genes is done. The p-values of the MR and MIMRMS methods are 0.2794 and 0.04412, respectively. However, in case of the mRMR method, not a single gene is overlapped with the LIST-1.

For the LIST-2, the results are reported in Table 6.2. While 11 genes selected by the MIMRMS method are related to colorectal cancer, only 5 and 3 colorectal cancer-related genes are identified by the MR and mRMR methods, respectively. Hence, the p-value obtained using the MIMRMS method is 4.452e-014 with respect to the LIST-2, while the MR and mRMR methods generate p-values 2.466e-12 and 0.0001762, respectively. For the LIST-2, the total number of genes analyzed in the study of Sabates-Bellver et al. [47] is 20827.

Finally, Table 6.3 represents the statistical significance test of overlapped genes of the MR, mRMR, and MIMRMS methods with respect to the genes of the LIST-3. In this case, the Fisher's exact test generates a lower p-value of 6.026e-006 for the MIMRMS method, which is significantly better than the p-values of 0.04794 and 0.0005489 obtained by the MR and mRMR methods, respectively. Total 21892

Table 6.3 Overlap with LIST-3

Methods		Yes	No	Total
MR	Yes	1	7	8
	No	133	21751	21884
mRMR	Yes	2	4	6
	No	132	21754	21886
MIMRMS	Yes	4	16	20
	No	130	21742	21872
	Total	134	21758	21892

genes are analyzed in the study of Nagaraj and Reverter [38]. All the results reported in Tables 6.1, 6.2, and 6.3 demonstrate that the overlap between the effective gene set $\mathbb{S}_{\text{GE}}$ obtained by the MIMRMS method and the three gene lists is significantly high as compared to both MR and mRMR methods.

6.3.4 PPI Data and Shortest Path Analysis

The PPI network is generated for each gene selected by three gene selection algorithms, namely, MR, mRMR, and MIMRMS. These networks are generated using the STRING database. The level of interaction between the selected set $\mathbb{S}_{\text{GE}}$ of genes and the proteins of the STRING database is measured by their confidence score. For the MIMRMS method, among the 20 genes of $\mathbb{S}_{\text{GE}}$, one gene, namely, **TMIGD**, does not have any interaction with any other genes; hence 19 networks are generated in this case using the STRING database. These 19 networks or subnetworks are further merged. The shortest path analysis is conducted on this merged PPI network. Total 342 shortest paths are calculated between each of the gene pairs of $\mathbb{S}_{\text{GE}}$ set generated by the MIMRMS method using the Dijkstra's algorithm.

Figure 6.3 shows the PPI network for 20 genes obtained by the MIMRMS method, along with their confidence scores. The nodes marked yellow represent the 20 genes of $\mathbb{S}_{\text{GE}}$ set identified by the MIMRMS method, while other 77 genes of $\mathbb{S}_{\text{PPI}}$ are existing in the shortest paths. The values on the edges represent the edge weights to quantify the interaction confidence. The smaller value indicates the stronger interaction between the two nodes. These 77 genes are further ranked according to their betweenness value. Among the 77 genes, **AR** has the largest betweenness value of 90, indicating that there are 90 shortest paths going through this gene. Accordingly, **AR** may play an important role to connect the 19 candidate genes and hence this gene may be related to colorectal cancer. Similarly, Figs. 6.4 and 6.5 represent the PPI networks for the MR and mRMR methods, respectively.

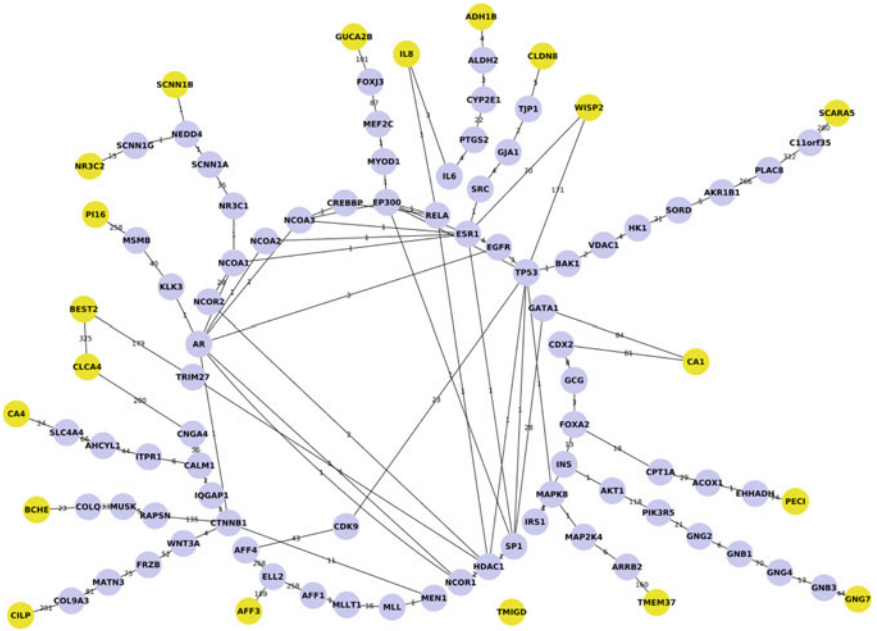

Fig. 6.3 PPI network for 20 genes obtained by the MIMRMS method, along with their confidence scores

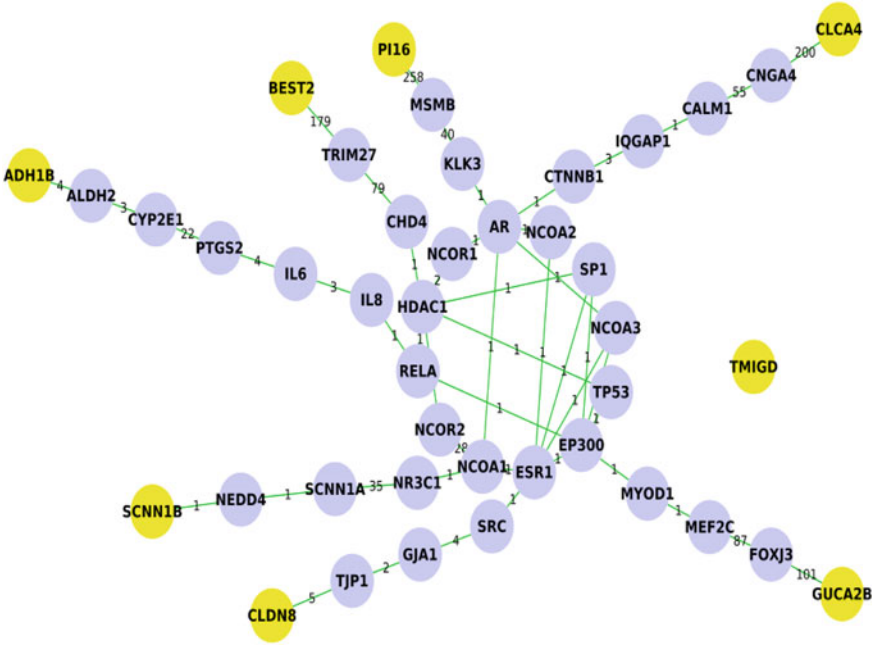

Fig. 6.4 PPI network for 8 genes obtained by the MR method, along with their confidence scores

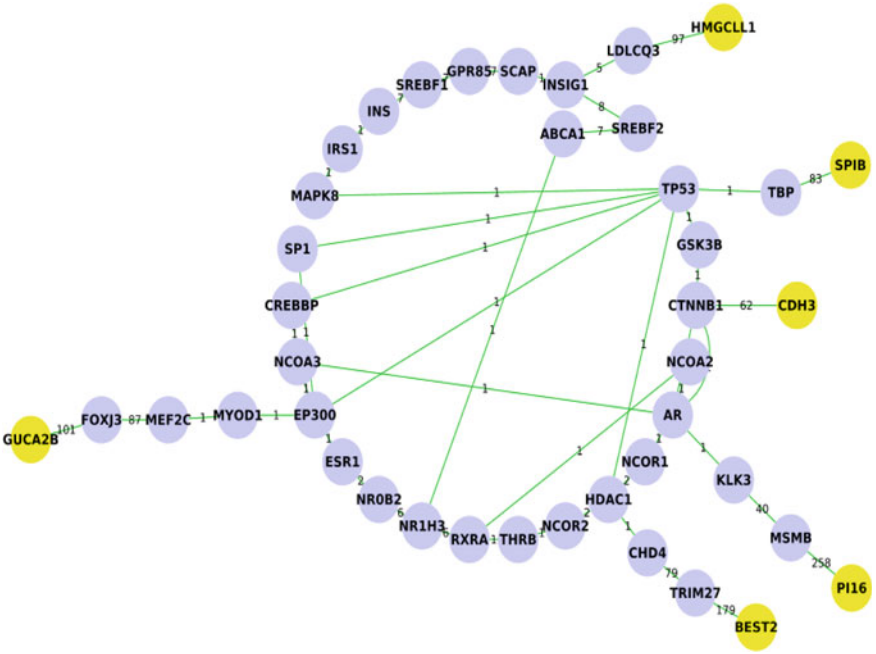

Fig. 6.5 PPI network for 6 genes obtained by the mRMR method, along with their confidence scores

6.3.5 Comparative Performance Analysis of Different Methods

This section presents the comparative performance analysis of the MR, mRMR, and MIMRMS methods, in terms of the statistical analysis. Statistical significance analysis of gene sets $\mathbb{S}_{GE+PPI}$ obtained by three approaches are discussed. The set $\mathbb{S}_{GE+PPI}$ generated by the MIMRMS method contains 97 genes, whereas that of both the MR and mRMR methods contains 41 genes. Tables 6.4, 6.5, and 6.6 represent the statistical significance analysis of the disease genes sets $\mathbb{S}_{GE+PPI}$ obtained by the MR, mRMR, and MIMRMS methods.

Table 6.4 represents the degree of overlapping of the gene sets $\mathbb{S}_{GE+PPI}$ generated by the MR, mRMR, MIMRMS methods with the genes of LIST-1. Out of 97 genes of the set $\mathbb{S}_{GE+PPI}$ generated by the MIMRMS method, 22 genes are overlapped with the LIST-1, so the corresponding p-value is 2.792e-11. On the other hand, out of 41 genes of $\mathbb{S}_{GE+PPI}$, 8 genes generated by both MR and mRMR methods are overlapped with the LIST-1, generating a p-value 0.0001908.

Table 6.5 depicts the overlapping of genes of the set $\mathbb{S}_{GE+PPI}$ generated by three gene selection methods with the genes of the LIST-2. Out of 97 genes of the set $\mathbb{S}_{GE+PPI}$ selected by the MIMRMS method, 15 genes are found to be overlapped with the LIST-2, and the corresponding p-value is 1.758e-09. On the other hand,

Table 6.4 Overlap with LIST-1

Methods		Yes	No	Total
MR	Yes	8	33	41
	No	734	17716	18450
mRMR	Yes	8	33	41
	No	734	17716	18450
MIMRMS	Yes	22	75	97
	No	720	17674	18394
	Total	742	17749	18491

Table 6.5 Overlap with LIST-2

Methods		Yes	No	Total
MR	Yes	7	34	41
	No	431	20355	20786
mRMR	Yes	4	37	41
	No	434	20352	20786
MIMRMS	Yes	15	82	97
	No	423	20307	20730
	Total	438	20389	20827

Table 6.6 Overlap with LIST-3

Methods		Yes	No	Total
MR	Yes	5	36	41
	No	129	21722	21851
mRMR	Yes	3	38	41
	No	131	21720	21851
MIMRMS	Yes	9	88	97
	No	125	21670	21795
	Total	134	21758	21892

only 7 and 4 genes of the set $\mathbb{S}_{\text{GE+PPI}}$ of the MR and mRMR methods, respectively, are found to be overlapped with the LIST-2. Hence, the corresponding p-values are 2.102e-05 and 0.01057 for the MR and mRMR methods, respectively. Finally, Table 6.6 shows the overlapping of genes of the set $\mathbb{S}_{\text{GE+PPI}}$ with the LIST-3. Out of 97 genes, 9 genes selected by the MIMRMS method are overlapped with the LIST-3, and the corresponding p-value is 8.338e-09. On the other hand, 5 and 3 genes of the MR and mRMR methods, respectively, overlap with the genes of the LIST-3, and generate p-values 5.005e-06 and 0.002017, respectively.

All the results reported in Tables 6.1, 6.2, 6.3, 6.4, 6.5, and 6.6 establish the fact that the MIMRMS method selects more number of disease-related genes than both MR and mRMR methods. The better performance of the MIMRMS algorithm over

mRMR algorithm is achieved due to the fact that the mRMR algorithm selects a subset of genes from the whole gene set by maximizing the relevance and minimizing the redundancy of the selected genes. The redundancy measure of the mRMR method does not take into account the supervised information of class labels, while both relevance and significance criteria of the MIMRMS method are computed based on the class labels. In effect, the MIMRMS method provides better performance than the mRMR method. Extensive experimental study on colorectal cancer also establishes the fact that the genes identified by the integrated method have more colorectal cancer genes than the genes identified from the gene expression profiles alone. All these results indicate that the integrated method is quite promising and may become a useful tool for identifying disease genes.

6.4 Conclusion and Discussion

The problem of identification of disease genes is addressed in this chapter. Several existing approaches are discussed, along with their merits and demerits. Next, a new approach is presented, integrating judiciously colorectal gene expression data and protein–protein interaction (PPI) data, to identify disease genes of colorectal cancer. First, a set of differentially expressed genes are selected using mutual information-based maximum relevance-maximum significance (MRMS) framework. The selected gene set is then used to construct a graph using the PPI data from the STRING database. Finally, the shortest path in the graph is identified and genes on these paths are considered as disease genes. The statistical analysis of the gene set is performed to observe whether the identified gene set is significantly related to colorectal cancer or not. A highly statistical significant set of colorectal cancer genes are selected by the new method compared to other related methods.

In this regard, it should be mentioned that the rough set-based MRMS (RSMRMS) method, reported in Chap. 4, can also be used to identify differentially expressed genes from microarray gene expression data sets [35, 44]. Next chapter presents the application of the RSMRMS method for selection of differentially expressed miRNAs from microarray data.

References

1. Althaus IW, Gonzales AJ, Chou JJ, Romero DL, Deibel MR, Chou KC, Kezdy FJ, Resnick L, Busso ME, So AG (1993) The quinoline U-78036 is a potent inhibitor of HIV-1 reverse transcriptase. J Biol Chem 268(20):14,875–14,880
2. Altshuler D, Daly MJ, Lander ES (2008) Genetic mapping in human disease. Science 322(5903):881–888
3. Andraos J (2008) Kinetic plasticity and the determination of product ratios for kinetic schemes leading to multiple products without rate laws—new methods based on directed graphs. Can J Chem 86(4):342–357

4. Barrenas F, Chavali S, Holme P, Mobini R, Benson M (2009) Network properties of complex human disease genes identified through genome-wide association studies. PLoS One 4(11):e8090
5. Bogdanov P, Singh A (2010) Molecular function prediction using neighborhood features. IEEE/ACM Trans Comput Biol Bioinform 7(2):208–217
6. Cai YD, Huang T, Feng KY, Hu L, Xie L (2010) A unified 35-gene signature for both subtype classification and survival prediction in diffuse large b-cell lymphomas. PLoS One 5(9):e12,726
7. Chen J, Aronow B, Jegga A (2009) Disease candidate gene identification and prioritization using protein interaction networks. BMC Bioinform 10(1):73
8. Chen L, Cai YD, Shi XH, Huang T (2010) Analysis of metabolic pathway using hybrid properties. PLoS One 5(6):e10,972
9. Chou KC (1990) Applications of graph theory to enzyme kinetics and protein folding kinetics: steady and non-steady-state systems. Biophys Chem 35(1):1–24
10. Chou KC (1993) Graphic rule for non-steady-state enzyme kinetics and protein folding kinetics. J Math Chem 12(1):97–108
11. Chou KC (2010) Graphic rule for drug metabolism systems. Curr Drug Metab 11:369–378
12. Chou KC, Forsen S (1980) Graphical rules for enzyme-catalysed rate laws. Biochem J 187:829–835
13. Chou KC, Kezdy FJ, Reusser F (1994) Kinetics of processive nucleic acid polymerases and nucleases. Anal Biochem 221(2):217–230
14. Dermitzakis ET (2008) From gene expression to disease risk. Nat Genet 40:492–493
15. Dijkstra EW (1959) A note on two problems in connexion with graphs. Numer Math 1(1):269–271
16. Ding C, Peng H (2003) Minimum redundancy feature selection from microarray gene expression data. In: Proceedings of the international conference on computational systems, bioinformatics, pp. 523–528
17. Ding C, Peng H (2005) Minimum redundancy feature selection from microarray gene expression data. J Bioinform Comput Biol 3(2):185–205
18. Duda RO, Hart PE, Stork DG (1999) Pattern classification and scene analysis. Wiley, New York
19. Goh KI, Cusick ME, Valle D, Childs B, Vidal M, Barabsi AL (2007) The human disease network. Proc Natl Acad Sci 104(21):8685–8690
20. Hinoue T, Weisenberger DJ, Lange CP, Shen H, Byun HM, Van Den Berg D, Malik S, Pan F, Noushmehr H, van Dijk CM, Tollenaar RAEM, Laird PW (2012) Genome-scale analysis of aberrant DNA methylation in colorectal cancer. Genome Res 22(2):271–282
21. Huang T, Cui W, Hu L, Feng K, Li YX, Cai YD (2009) Prediction of pharmacological and xenobiotic responses to drugs based on time course gene expression profiles. PLoS One 4(12):e8126
22. Huang T, Cai YD, Chen L, Hu LL, Kong XY, Li YX, Chou KC (2010) Selection of reprogramming factors of induced pluripotent stem cells based on the protein interaction network and functional profiles. PLoS One 5(9):e12,726
23. Huang T, Shi XH, Wang P, He Z, Feng KY, Hu L, Kong X, Li YX, Cai YD, Chou KC (2010) Analysis and prediction of the metabolic stability of proteins based on their sequential features, subcellular locations and interaction networks. PLoS One 5(6):e10,972
24. Huang T, Chen L, Cai YD, Chou KC (2011) Classification and analysis of regulatory pathways using graph property, biochemical and physicochemical property, and functional property. PLoS One 6(9):e25,297
25. Huret JL, Dessen P, Bernheim A (2003) Atlas of genetics and cytogenetics in oncology and haematology, year 2003. Nucleic Acids Res 31(1):272–274
26. Jia P, Zheng S, Long J, Zheng W, Zhao Z (2011) dmGWAS: dense module searching for genome-wide association studies in protein–protein interaction networks. Bioinformatics 27(1):95–102
27. Karaoz U, Murali TM, Letovsky S, Zheng Y, Ding C, Cantor CR, Kasif S (2004) Whole-genome annotation by using evidence integration in functional-linkage networks. Proc Natl Acad Sci 101(9):2888–2893

28. Karni S, Soreq H, Sharan R (2009) A network-based method for predicting disease-causing genes. J Comput Biol 16(2):181–189
29. Keshava Prasad TS, Goel R, Kandasamy K, Keerthikumar S, Kumar S, Mathivanan S, Telikicherla D, Raju R, Shafreen B, Venugopal A, Balakrishnan L, Marimuthu A, Banerjee S, Somanathan DS, Sebastian A, Rani S, Ray S, Harrys Kishore CJ, Kanth S, Ahmed M, Kashyap MK, Mohmood R, Ramachandra YL, Krishna V, Rahiman BA, Mohan S, Ranganathan P, Ramabadran S, Chaerkady R, Pandey A (2009) Human protein reference database-2009 update. Nucleic Acids Res 37(suppl 1):D767–D772
30. Kohler S, Bauer S, Horn D, Robinson PN (2008) Walking the interactome for prioritization of candidate disease genes. Am J Hum Genet 82(4):949–958
31. Kourmpetis YAI, van Dijk ADJ, Bink MCAM, van Ham RCHJ, ter Braak CJF (2010) Bayesian Markov random field analysis for protein function prediction based on network data. PLoS One 5(2):e9293
32. Letovsky S, Kasif S (2003) Predicting protein function from protein/protein interaction data: a probabilistic approach. Bioinformatics 19(suppl 1):i197–i204
33. Li BQ, Huang T, Liu L, Cai YD, Chou KC (2012) Identification of colorectal cancer related genes with mRMR and shortest path in protein–protein interaction network. PLoS one 7(4):e33,393
34. Li Y, Li J (2012) Disease gene identification by random walk on multigraphs merging heterogeneous genomic and phenotype data. BMC Genomics 13(Suppl 7):S27
35. Maji P, Paul S (2011) Rough set based maximum relevance-maximum significance criterion and gene selection from microarray data. Int J Approximate Reasoning 52(3):408–426
36. Meltzer PS (2001) Spotting the target: microarrays for disease gene discovery. Curr Opin Genet Dev. 11(3):258–263
37. Mohammadi A, Saraee M, Salehi M (2011) Identification of disease-causing genes using microarray data mining and gene ontology. BMC Med Genomics 4(1):12
38. Nagaraj S, Reverter A (2011) A boolean-based systems biology approach to predict novel genes associated with cancer: application to colorectal cancer. BMC Syst Biol 5(1):35
39. Navlakha S, Kingsford C (2010) The power of protein interaction networks for associating genes with diseases. Bioinformatics 26(8):1057–1063
40. Ng KL, Ciou JS, Huang CH (2010) Prediction of protein functions based on function–function correlation relations. Comput Biol Med 40(3):300–305
41. Nitsch D, Tranchevent LC, Thienpont B, Thorrez L, Van Esch H, Devriendt K, Moreau Y (2009) Network analysis of differential expression for the identification of disease-causing genes. PLoS One 4(5):e5526
42. Novershtern N, Itzhaki Z, Manor O, Friedman N, Kaminski N (2008) A functional and regulatory map of asthma. Am J Resp Cell Mol Biol 38(3):324–336
43. Oti M, Snel B, Huynen MA, Brunner HG (2006) Predicting disease genes using protein–protein interactions. J Med Genet 43(8):691–698
44. Paul S, Maji P (2013) Gene ontology based quantitative index to select functionally diverse genes. Int J Mach Learn Cybern. doi:10.1007/s13042-012-0133-5.
45. Quenouille MH (1949) Approximate tests of correlation in time-series. J Roy Stat Soc Ser B (Methodol) 11(1):68–84
46. Ruan X, Wang J, Li H, Perozzi RE, Perozzi EF (2008) The use of logic relationships to model colon cancer gene expression networks with mRNA microarray data. J Biomed Inform 41(4):530–543
47. Sabates-Bellver J, Van der Flier LG, de Palo M, Cattaneo E, Maake C, Rehrauer H, Laczko E, Kurowski MA, Bujnicki JM, Menigatti M, Luz J, Ranalli TV, Gomes V, Pastorelli A, Faggiani R, Anti M, Jiricny J, Clevers H, Marra G (2007) Transcriptome profile of human colorectal adenomas. Mol Cancer Res 5(12):1263–1275
48. Sharan R, Ulitsky I, Shamir R (2007) Network-based prediction of protein function. Mol Syst Biol 3(88):1–13
49. Szklarczyk D, Franceschini A, Kuhn M, Simonovic M, Roth A, Minguez P, Doerks T, Stark M, Muller J, Bork P, Jensen LJ, von Mering C (2011) The STRING database in 2011: functional

interaction networks of proteins, globally integrated and scored. Nucleic Acids Res 39(suppl 1):D561–D568
50. Vajda I (1989) Theory of statistical inference and information. Kluwer Academic, Dordrecht
51. Wu C, Zhu J, Zhang X (2012) Integrating gene expression and protein–protein interaction network to prioritize cancer-associated genes. BMC Bioinform 13(1):182
52. Zhao J, Jiang P, Zhang W (2010) Molecular networks for the study of TCM pharmacology. Briefings Bioinform 11(4):417–430
53. Zhao J, Yang TH, Huang Y, Holme P (2011) Ranking candidate disease Genes from gene expression and protein interaction: a Katz-centrality based approach. PLoS One 6(9):e24,306
54. Zhou GP (2011) The disposition of the LZCC protein residues in Wenxiang diagram provides new insights into the protein–protein interaction mechanism. J Theor Biol 284(1):142–148
55. Zhou GP, Deng MH (1984) An extension of Chou's graphic rules for deriving enzyme kinetic equations to systems involving parallel reaction pathways. Biochem J 222:169–176

Chapter 7
Rough Sets for Insilico Identification of Differentially Expressed miRNAs

7.1 Introduction

The microRNAs or miRNAs, a class of short approximately 22-nucleotide non-coding RNAs found in many plants and animals, often act posttranscriptionally to inhibit gene or mRNA expression. Hence, the miRNAs are related to diverse cellular processes and regarded as important components of the gene regulatory network. Multiple reports have noted the utility of miRNAs for the diagnosis of cancer and other diseases. Unlike with mRNAs, a modest number of miRNAs, 200 in total, might be sufficient to classify human cancers [16]. Moreover, the bead-based miRNA detection method has the attractive property of being not only accurate and specific, but also easy to implement in a routine clinical setting. In addition, unlike mRNAs, miRNAs remain largely intact in routinely collected, formalin-fixed, paraffin-embedded clinical tissues [4]. Recent studies have also shown that miRNAs can be detected in serum. These studies offer the promise of utilizing miRNA screening via less invasive blood-based mechanisms. In addition, mature miRNAs are relatively stable. These phenomena make miRNAs superior molecular markers and targets for interrogation and as such, miRNA expression profiling can be utilized as a tool for cancer diagnosis and other diseases.

The functions of miRNAs appear to be different in various cellular functions. Just as miRNA is involved in the normal functioning of eukaryotic cells, so has dysregulation of miRNA been associated with disease [14]. It indicates that the miRNAs can prove to be potential biomarkers for developing a diagnostic tool. Hence, insilico identification of differentially expressed miRNAs that target genes involved in diseases is necessary. These differentially expressed miRNAs can be further used in developing effective diagnostic tools. Recently, few studies are carried out to identify differentially expressed miRNAs [3, 5, 9, 38, 45]. However, absence of robust method makes it an open problem. Hence, the data sets are needed to be explored for understanding the complex biological activities of miRNAs.

P. Maji and S. Paul, *Scalable Pattern Recognition Algorithms*,
DOI: 10.1007/978-3-319-05630-2_7,

A miRNA expression data set can be represented by an expression table or matrix, where each row corresponds to one particular miRNA, each column to a sample, and each entry of the matrix is the measured expression level of a particular miRNA in a sample, respectively. However, for microarray data, the number of training samples is typically very small, while the number of miRNAs is in the thousands. Hence, the prediction rule formed by support vector machine or any other classifier may not be able to be formed by using all available miRNAs. Even if all the miRNAs can be used, the use of all the miRNAs allows the noise associated with miRNAs of little or no discriminatory power, which inhibits and degrades the performance of the prediction rule in its application to unclassified or test samples. In other words, although the apparent error rate, which is the proportion of the training samples misclassified by the prediction rule, will decrease as it is formed from more and more miRNAs, its error rate in classifying samples outside the training set eventually will increase. That is, the generalization error of the prediction rule will be increased if it is formed from a sufficiently large number of miRNAs. Hence, in practice, consideration has to be given to implement some procedure of feature selection for reducing the number of miRNAs to be used in constructing the prediction rule [1].

The method called significance analysis of microarrays is used in several works [10, 15, 27, 28, 36, 37] to identify differentially expressed miRNAs. Different statistical tests are also employed to identify differentially expressed miRNAs [2, 3, 5, 9, 16, 26, 38, 42, 45, 46]. Rui et al. [43] used particle swarm optimization technique for selecting important miRNAs that contribute to the discrimination of different cancer types. Different feature selection algorithms such as mutual information [6] or f-information [17] based minimum redundancy-maximum relevance framework , reported in Chap. 5, can also be used to select a set of nonredundant and relevant miRNAs for sample classification. A detailed survey on several feature selection algorithms is reported in Chap. 4.

One of the main problems in miRNA expression data analysis is uncertainty. Some of the sources of this uncertainty include imprecision in computations and vagueness in class definition. In this background, the rough set theory has gained popularity in modeling and propagating uncertainty. It deals with vagueness and incompleteness and is proposed for indiscernibility in classification according to some similarity [35]. A brief survey on different rough set-based feature selection algorithms is reported in Chap. 4. The theory of rough sets has also been successfully applied to microarray data analysis in [8, 18, 21, 23–25, 31, 32, 39, 40].

In general, the performance of the prediction rule generated by a classifier for a subset of selected miRNAs is evaluated by leave-one-out cross-validation (LOOCV) error. Given that the entire set of available samples is relatively small, in practice, one would like to make full use of all available samples in the miRNA selection and training of the prediction rule. But, if the LOOCV is calculated within the miRNA selection process, there is a selection bias in it when it is used as an estimate of the prediction error. The LOOCV error of the prediction rule obtained during the selection of the miRNAs provides a too optimistic estimate of the prediction error rate. Hence, an external cross-validation should be undertaken subsequent to the

miRNA selection process to correct for this selection bias. Alternatively, the bootstrap procedure can be used [7, 19].

Although, the LOOCV error with external cross-validation is nearly unbiased, it can be highly variable in the sense that there is no guarantee that the same subset of miRNAs will be obtained as during the original training of the rule on all the training samples. Indeed, with the huge number of miRNAs available, it generally will yield a subset of miRNAs that has at most only a few miRNAs in common with the subset selected during the original training of the rule. Suitably defined bootstrap procedures can reduce the variability of the LOOCV error in addition to providing a direct assessment of variability for estimated parameters in the prediction rule. However, the bootstrap approach overestimates the error. To reduce the weakness of both these approaches, Efron and Tibshirani introduced the concept of $B.632+$ error for correcting the upward bias in bootstrap error with the downwardly biased apparent error [7], which is very much applicable for the data sets with small number of training samples and large number of miRNAs.

In this regard, this chapter presents a novel approach, proposed by Paul and Maji in [34], for insilico identification of differentially expressed miRNAs from expression data sets. It integrates the merit of rough set-based feature selection algorithm using maximum relevance-maximum significance criterion (RSMRMS), reported in Chap. 4, and the concept of so-called $B.632+$ error rate [7]. The RSMRMS algorithm selects a subset of miRNAs from a data set by maximizing both relevance and significance of the selected miRNAs. It employs rough set theory to compute both relevance and significance of the miRNAs. Hence, the only information required in the feature selection method is in the form of equivalence partitions for each miRNA, which can be automatically derived from the given microarray data set. A fuzzy set-based discretization method is presented to generate equivalence classes required to compute both relevance and significance of miRNAs using rough set theory. This avoids the need for domain experts to provide information on the data involved and ties in with the advantage of rough sets is that it requires no information other than the data set itself. On the other hand, the $B.632+$ error rate minimizes the variability and biasedness of the derived results. The support vector machine is used to compute the $B.632+$ error rate as well as several other types of error rates as it maximizes the margin between data samples in different classes. The effectiveness of the new approach, along with a comparison with other related approaches, is demonstrated on a set of miRNA expression data sets.

The chapter is organized as follows: Sect. 7.2 presents the miRNA selection method reported in [34], which covers the basics of the RSMRMS algorithm, and the concepts of fuzzy discretization and $B.632+$ error rate. Implementation details, a brief description of several miRNA data sets used in this study, experimental results, and a comparison among different algorithms are presented in Sect. 7.3. Concluding remarks are given in Sect. 7.4.

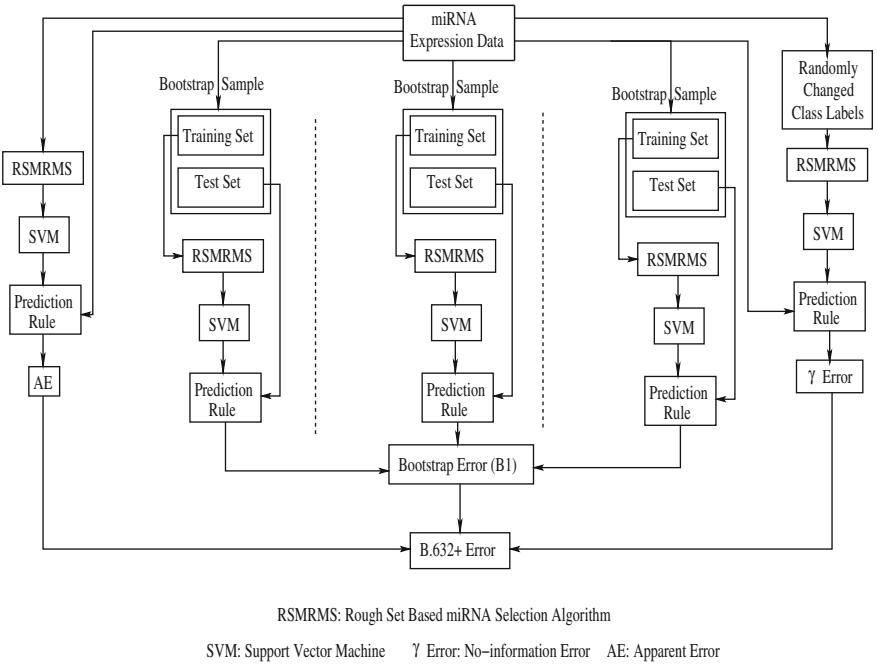

Fig. 7.1 Schematic flow diagram of the insilico approach for identification of differentially expressed miRNAs

7.2 Selection of Differentially Expressed miRNAs

The rough set-based insilico approach is illustrated in Fig. 7.1. It mainly consists of rough set-based feature selection method (RSMRMS) described in Chap. 4, support vector machine (SVM) [41], and several types of error analysis parts, namely, apparent error (AE), bootstrap error ($B1$), no-information error (γ), and $B.632+$ error. The RSMRMS algorithm selects a set of miRNAs from a given miRNA expression data. The selected set of miRNAs is then used to design the SVM classifier, and the effectiveness of the build up SVM classifier is further tested by using unseen data. In order to calculate $B.632+$ error, at first, apparent error (AE) is calculated. This error is generated, when the same data set is used to train and test a classifier. Next, $B1$ error is calculated from k bootstrap samples. Finally, by randomly perturbing the class label of a given data set, no-information error (γ) is calculated. The mutated data set is used for miRNA selection and the generated set of miRNAs is used to build the SVM. Then, the trained SVM is tested using the original data set. The error generated by this procedure is known as no-information error (γ). Using apparent error (AE), $B1$ error, and γ error, lastly $B.632+$ error is calculated. The RSMRMS method is discussed in Chap. 4, while a brief introduction of the SVM is reported in

Chaps. 3 and 4. Hence, this section presents only the concepts of fuzzy equivalence classes used to generate equivalence classes for rough sets and different types of errors, along with a brief overview of the RSMRMS algorithm.

7.2.1 RSMRMS Algorithm

In real data analysis such as microarray data, the data set may contain a number of insignificant features. The presence of such irrelevant and insignificant features may lead to a reduction in the useful information. Ideally, the selected features should have high relevance with the classes and high significance in the feature set. The features with high relevance are expected to be able to predict the classes of the samples. However, if insignificant features are present in the subset, they may reduce the prediction capability and may contain similar biological information. A feature set with high relevance and high significance enhances the predictive capability. Accordingly, a measure is required that can enhance the effectiveness of feature set. In this work, the rough set theory is used to select the relevant and significant miRNAs from high dimensional microarray data sets.

Let $\mathbb{C} = \{\mathscr{A}_1, \cdots, \mathscr{A}_i, \cdots, \mathscr{A}_j, \cdots, \mathscr{A}_m\}$ be the set of m miRNAs of a given microarray data set and $\mathbb{S}$ is the set of selected miRNAs. Define $\gamma_{\mathscr{A}_i}(\mathbb{D})$ as the relevance of the miRNA $\mathscr{A}_i$ with respect to the class labels $\mathbb{D}$ while $\sigma_{\{\mathscr{A}_i,\mathscr{A}_j\}}(\mathbb{D}, \mathscr{A}_j)$ as the significance of the miRNA $\mathscr{A}_j$ with respect to the set $\{\mathscr{A}_i, \mathscr{A}_j\}$. The total relevance of all selected miRNAs is as follows:

$$\mathscr{J}_{\text{relev}} = \sum_{\mathscr{A}_i \in \mathbb{S}} \gamma_{\mathscr{A}_i}(\mathbb{D}) \tag{7.1}$$

while the total significance among the selected miRNAs is

$$\mathscr{J}_{\text{signf}} = \sum_{\mathscr{A}_i \neq \mathscr{A}_j \in \mathbb{S}} \sigma_{\{\mathscr{A}_i,\mathscr{A}_j\}}(\mathbb{D}, \mathscr{A}_j). \tag{7.2}$$

Therefore, the problem of selecting a set $\mathbb{S}$ of relevant and significant miRNAs from the whole set $\mathbb{C}$ of m miRNAs is equivalent to maximize both $\mathscr{J}_{\text{relev}}$ and $\mathscr{J}_{\text{signf}}$, that is, to maximize the objective function $\mathscr{J}$, where

$$\mathscr{J} = \mathscr{J}_{\text{relev}} + \beta\, \mathscr{J}_{\text{signf}} \tag{7.3}$$

that is,

$$\mathscr{J} = \sum_{\mathscr{A}_i \in \mathbb{S}} \gamma_{\mathscr{A}_i}(\mathbb{D}) + \beta \sum_{\mathscr{A}_i \neq \mathscr{A}_j \in \mathbb{S}} \sigma_{\{\mathscr{A}_i,\mathscr{A}_j\}}(\mathbb{D}, \mathscr{A}_j) \tag{7.4}$$

where β is a weight parameter. To solve the above problem, a greedy algorithm is used in [24]. The relevance and significance of a miRNA are calculated based on the theory of rough sets using (4.6) and (4.7), respectively. The weight parameter β in the rough set-based MRMS (RSMRMS) algorithm regulates the relative importance of the significance of the candidate miRNA with respect to the already-selected miRNAs and the relevance with the output class. If β is zero, only the relevance with the output class is considered for each miRNA selection. If β increases, this measure is incremented by a quantity proportional to the total significance with respect to the already-selected miRNAs. The presence of a β value larger than zero is crucial in order to obtain good results. If the significance between miRNAs is not taken into account, selecting the miRNAs with the highest relevance with respect to the output class may tend to produce a set of redundant miRNAs that may leave out useful complementary information. Details of the RSMRMS algorithm are available in Chap. 4.

7.2.2 Fuzzy Discretization

In miRNA expression data, the class labels of samples are represented by discrete symbols, while the expression values of miRNAs are continuous. Hence, to measure both relevance and significance of miRNAs using rough set theory, the continuous expression values of a miRNA have to be divided into several discrete partitions to generate equivalence classes. In this regard, a fuzzy set-based discretization method is used to generate equivalence classes required to compute both relevance and significance of the miRNAs.

Fuzzy set was introduced by Zadeh [44] as a generalization of the classical set theory. A fuzzy set A in a space of objects $\mathbb{U} = \{x_i\}$ is a class of events with a continuum of grades of membership and is characterized by a membership function $\mu_A(x_i)$ that associates with each element in $\mathbb{U}$ a real number in the interval [0, 1] with the value of $\mu_A(x_i)$ at x_i representing the grade of membership of x_i in A. Formally, a fuzzy set A with its finite number of supports $x_1, \cdots, x_i, \cdots, x_n$ is defined as a collection of ordered pairs $A = \{\mu_A(x_i)/x_i, i = 1, \cdots, n\}$, where the support of A is an ordinary subset of $\mathbb{U}$ and is defined as

$$S(A) = \{x_i | x_i \in \mathbb{U} \text{ and } \mu_A(x_i) > 0\}. \tag{7.5}$$

Here $\mu_A(x_i)$ represents the degree to which an object x_i may be a member of A or belong to A. If the support of a fuzzy set is only a single object $x_1 \in \mathbb{U}$, then $A = \mu_A(x_1)/x_1$ is called a fuzzy singleton. Hence, if $\mu_A(x_1) = 1$, $A = \frac{1}{x_1}$ denotes a nonfuzzy singleton. In terms of the constituent singletons, the fuzzy set A with its finite number of supports $x_1, \cdots, x_i, \cdots, x_n$ can also be expressed in union form as

$$A = \{\mu_A(x_1)/x_1 + \cdots + \mu_A(x_i)/x_i + \cdots + \mu_A(x_n)/x_n\} \tag{7.6}$$

where the sign + denotes the union [13]. Assignment of membership functions of a fuzzy subset is subjective in nature, and reflects the context in which the problem is viewed.

The family of normal fuzzy sets produced by a fuzzy partitioning of the universe of discourse can play the role of fuzzy equivalence classes. Given a finite set $\mathbb{U}$, $\mathbb{C}$ is a fuzzy condition attribute set in $\mathbb{U}$, which generates a fuzzy equivalence partition on $\mathbb{U}$. If c denotes the number of fuzzy equivalence classes generated by the fuzzy equivalence relation and n is the number of objects in $\mathbb{U}$, then c-partitions of $\mathbb{U}$ are sets of (cn) values $\{\mu_{ij}^{\mathbb{C}}\}$ that can be conveniently arrayed as a $(c \times n)$ matrix $\mathbb{M}_{\mathbb{C}} = [\mu_{ij}^{\mathbb{C}}]$, which is denoted by

$$\mathbb{M}_{\mathbb{C}} = \begin{pmatrix} \mu_{11}^{\mathbb{C}} & \mu_{12}^{\mathbb{C}} & \cdots & \mu_{1n}^{\mathbb{C}} \\ \mu_{21}^{\mathbb{C}} & \mu_{22}^{\mathbb{C}} & \cdots & \mu_{2n}^{\mathbb{C}} \\ \cdots & \cdots & \cdots & \cdots \\ \mu_{c1}^{\mathbb{C}} & \mu_{c2}^{\mathbb{C}} & \cdots & \mu_{cn}^{\mathbb{C}} \end{pmatrix} \tag{7.7}$$

where $\mu_{ij}^{\mathbb{C}} \in [0, 1]$ represents the membership of object x_j in the ith fuzzy equivalence partition or class F_i [20, 21].

Each row of the matrix $\mathbb{M}_{\mathbb{C}}$ is a fuzzy equivalence partition or class. In the rough set-based feature selection method, the π function in one dimensional form is used to assign membership values to different fuzzy equivalence classes for the input miRNAs. A fuzzy set with membership function $\pi(x; \bar{c}, \sigma)$ represents a set of points clustered around $\bar{c}$, where

$$\pi(x; \bar{c}, \sigma) = \begin{cases} 2(1 - \frac{||x-\bar{c}||}{\sigma})^2 & \text{for } \frac{\sigma}{2} \leq ||x - \bar{c}|| \leq \sigma \\ 1 - 2(\frac{||x-\bar{c}||}{\sigma})^2 & \text{for } 0 \leq ||x - \bar{c}|| \leq \frac{\sigma}{2} \\ 0 & \text{otherwise} \end{cases} \tag{7.8}$$

where $\sigma > 0$ is the radius of the π function with $\bar{c}$ as the central point and $||\cdot||$ denotes the Euclidean norm. When the pattern x lies at the central point $\bar{c}$ of a class, then $||x - \bar{c}|| = 0$ and its membership value is maximum, that is, $\pi(\bar{c}; \bar{c}, \sigma) = 1$. The membership value of a point decreases as its distance from the central point $\bar{c}$, that is, $||x - \bar{c}||$ increases. When $||x - \bar{c}|| = (\frac{\sigma}{2})$, the membership value of x is 0.5 and this is called a crossover point [30]. The $(c \times n)$ matrix $\mathbb{M}_{\mathscr{A}_i}$, corresponding to the ith miRNA $\mathscr{A}_i$, can be calculated from the c-fuzzy equivalence classes of the objects $x = \{x_1, \cdots, x_j, \cdots, x_n\}$, where

$$\mu_{kj}^{\mathscr{A}_i} = \frac{\pi(x_j; \bar{c}_k, \sigma_k)}{\sum_{l=1}^{c} \pi(x_j; \bar{c}_l, \sigma_l)}. \tag{7.9}$$

In effect, each position $\mu_{kj}^{\mathscr{A}_i}$ of the matrix $\mathbb{M}_{\mathscr{A}_i}$ must satisfy the following conditions:

$$\mu_{kj}^{\mathscr{A}_i} \in [0, 1]; \ \sum_{k=1}^{c} \mu_{kj}^{\mathscr{A}_i} = 1, \forall j \text{ and for any value of } k,$$
$$\text{if } s = \arg \max_j \{\mu_{kj}^{\mathscr{A}_i}\}, \text{ then } \max_j \{\mu_{kj}^{\mathscr{A}_i}\} = \max_l \{\mu_{ls}^{\mathscr{A}_i}\} > 0.$$

After the generation of the matrix $\mathbb{M}_{\mathscr{A}_i}$ corresponding to the miRNA $\mathscr{A}_i$, the object x_j is assigned to one of the c equivalence classes based on the maximum value of memberships of the object in different equivalence classes that follows next:

$$x_j \in F_p, \qquad \text{where } p = \arg \max_k \{\mu_{kj}^{\mathscr{A}_i}\}. \tag{7.10}$$

Each input real-valued miRNA in quantitative form can be assigned to different fuzzy equivalence classes in terms of membership values using the π fuzzy set with appropriate $\bar{c}$ and σ. The centers and radii of the π functions along each miRNA axis are determined automatically from the distribution of the training patterns. In the RSM-RMS algorithm, three fuzzy equivalence classes ($c = 3$), namely, low, medium, and high are considered. These three equivalence classes correspond to under-expression, baseline, and over-expression of continuous valued miRNAs, respectively. Corresponding to three fuzzy sets low, medium, and high, the following relations hold:

$$\bar{c}_1 = \bar{c}_{\text{low}}(\mathscr{A}_i); \ \bar{c}_2 = \bar{c}_{\text{medium}}(\mathscr{A}_i); \ \bar{c}_3 = \bar{c}_{\text{high}}(\mathscr{A}_i); \tag{7.11}$$

$$\sigma_1 = \sigma_{\text{low}}(\mathscr{A}_i); \ \sigma_2 = \sigma_{\text{medium}}(\mathscr{A}_i); \ \sigma_3 = \sigma_{\text{high}}(\mathscr{A}_i). \tag{7.12}$$

The parameters $\bar{c}$ and σ of each π fuzzy set are computed according to the following procedure [29]. Let $\bar{m}_i$ be the mean of the objects $x = \{x_1, \cdots, x_j, \cdots, x_n\}$ along the ith miRNA $\mathscr{A}_i$. Then $\bar{m}_{i_l}$ and $\bar{m}_{i_h}$ are defined as the mean along the ith miRNA of the objects having co-ordinate values in the range $[\mathscr{A}_{i_{\min}}, \bar{m}_i)$ and $(\bar{m}_i, \mathscr{A}_{i_{\max}}]$, respectively, where $\mathscr{A}_{i_{\max}}$ and $\mathscr{A}_{i_{\min}}$ denote the upper and lower bounds of the dynamic range of miRNA $\mathscr{A}_i$ for the training set. For three fuzzy sets low, medium, and high, the centers and corresponding radii are computed as follows:

$$\bar{c}_{\text{low}}(\mathscr{A}_i) = \bar{m}_{i_l}; \ \bar{c}_{\text{medium}}(\mathscr{A}_i) = \bar{m}_i; \ \bar{c}_{\text{high}}(\mathscr{A}_i) = \bar{m}_{i_h} \tag{7.13}$$

$$\sigma_{\text{low}}(\mathscr{A}_i) = 2(\bar{c}_{\text{medium}}(\mathscr{A}_i) - \bar{c}_{\text{low}}(\mathscr{A}_i)); \tag{7.14}$$

$$\sigma_{\text{high}}(\mathscr{A}_i) = 2(\bar{c}_{\text{high}}(\mathscr{A}_i) - \bar{c}_{\text{medium}}(\mathscr{A}_i)); \tag{7.15}$$

$$\sigma_{\text{medium}}(\mathscr{A}_i) = \eta \times \frac{\text{A}}{\text{B}}; \tag{7.16}$$

$$\mathrm{A} = \sigma_{\mathrm{low}}(\mathscr{A}_i)(\mathscr{A}_{i_{\max}} - c_{\mathrm{medium}}(\mathscr{A}_i)) + \sigma_{\mathrm{high}}(\mathscr{A}_i)(c_{\mathrm{medium}}(\mathscr{A}_i) - \mathscr{A}_{i_{\min}}); \quad (7.17)$$

and

$$\mathrm{B} = \{\mathscr{A}_{i_{\max}} - \mathscr{A}_{i_{\min}}\} \quad (7.18)$$

where η is a multiplicative parameter controlling the extent of the overlapping. The distribution of the patterns or objects along each miRNA axis is taken into account, while computing the corresponding centers and radii of the fuzzy sets. Also, the amount of overlap between the three fuzzy sets can be different along the different axis, depending on the distribution of the objects or patterns.

7.2.3 B.632+ Error Rate

In order to minimize variability and biasedness of derived result, the so-called $B.632+$ bootstrap approach [7] is used, which is defined as follows:

$$B.632+ = (1-\omega)AE + \omega B1 \quad (7.19)$$

where AE denotes the proportion of the original training samples misclassified, termed as apparent error rate, and $B1$ is the bootstrap error, defined as follows:

$$B1 = \frac{1}{n}\sum_{j=1}^{n}\frac{\sum_{k=1}^{\mathrm{M}} I_{jk}Q_{jk}}{\sum_{k=1}^{\mathrm{M}} I_{jk}} \quad (7.20)$$

where n is the number of original samples and M is the number of bootstrap samples. If the sample x_j is not contained in the kth bootstrap sample, then $I_{jk} = 1$, otherwise 0. Similarly, if x_j is misclassified, $Q_{jk} = 1$, otherwise 0. The weight parameter ω is given by

$$\omega = \frac{0.632}{1 - 0.368r}; \quad (7.21)$$

where

$$r = \frac{B1 - AE}{\gamma - AE}; \quad (7.22)$$

and

$$\gamma = \sum_{i=1}^{\mathrm{K}} p_i(1 - q_i); \quad (7.23)$$

where K is the number of classes, p_i is the proportion of the samples from the ith class, and q_i is the proportion of them assigned to the ith class. Also, γ is termed as the no-information error rate that would apply if the distribution of the class-membership label of the sample x_j did not depend on its feature vector.

7.3 Experimental Results

In this section, the performance of the rough sets-based maximum relevance-maximum significance (RSMRMS) algorithm is compared with that of mutual information-based minimum redundancy-maximum relevance (mRMR) algorithm [6] on three miRNA microarray data sets. The fuzzy set-based discretization method is also compared with several other discretization methods [17, 22]. The source code of the RSMRMS algorithm and fuzzy discretization method, written in C language, is available at http://www.isical.ac.in/~bibl/results/rsmrms/rsmrms.html. The margin classifier support vector machine (SVM) [41] is used to evaluate the performance of different algorithms. To compute different types of error rates obtained using the SVM, bootstrap approach is performed on each miRNA expression data set. For each training set, a set of differential miRNAs is first generated, and then the SVM is trained with the selected miRNAs. After the training, the information of miRNAs those were selected for the training set is used to generate test set and then the class label of the test sample is predicted using the SVM. For each data set, 50 top-ranked miRNAs are selected for the analysis. Each data set is preprocessed by standardizing each sample to zero mean and unit variance.

7.3.1 Data Sets Used

In this chapter, publicly available three miRNA expression data sets are used to establish the effectiveness of the approach. Three miRNA expression data sets with accession number GSE17681, GSE17846, and GSE29352 are downloaded from Gene Expression Omnibus (www.ncbi.nlm.nih.gov/geo/).

7.3.1.1 GSE17681

This data set has been generated to detect specific patterns of miRNAs in peripheral blood samples of lung cancer patients. As controls, blood of donors without known affection have been tested. The number of miRNAs, samples, and classes in this data set are 866, 36, and 2, respectively [12].

7.3.1.2 GSE17846

This data set represents the analysis of miRNA profiling in peripheral blood samples of multiple sclerosis and in blood of normal donors. It contains 864 miRNAs, 41 samples, and 2 classes [11].

7.3.1.3 GSE29352

In this data set, miRNA expression profiles in pancreatic cystic tumors with low malignant potential (serous microcystic adenomas) and high malignant potential (mucinous cystadenoma and intraductal papillary mucinous neoplasm (IPMN)) have been generated. These expression profiles are further compared in pancreatic ductal adenocarcinoma and carcinoma-ex-IPMN. It contains 43 samples, 885 miRNAs, and 3 classes.

7.3.2 Optimum Values of Different Parameters

The rough set-based miRNA selection algorithm uses weight parameter β to control the relative importance of significance of a miRNA with respect to its relevance. On the other hand, the multiplicative parameter η controls the degree of overlapping between the three fuzzy sets those are used to generate fuzzy equivalence classes. Hence, the performance of the RSMRMS approach very much depends on both the parameters β and η.

The value of β is varied from 0.0 to 1.0, while the parameter η varies from 0.5 to 2.0. Extensive experimental results are obtained for all values of β and η on three miRNA expression data sets. Figure 7.2 presents the variation of the $B.632+$ error rate obtained using the RSMRMS algorithm for different values of β and η on three miRNA data sets. From the results reported in Fig. 7.2, it is seen that as the value of β increases, the $B.632+$ error of the SVM decreases.

On the other hand, the error rate increases for very high or very low values of η. Table 7.1 presents the optimum values of β and η for which minimal $B.632+$ error rate of the SVM is achieved. From the results reported in Table 7.1, it is seen that the RSMRMS algorithm with $\beta \neq 0.0$ provides better result than that of $\beta = 0.0$ in all three cases, which justifies the importance of both relevance and significance criteria. The corresponding values of η indicate that very large or very small amounts of overlapping among the three fuzzy equivalence classes of input miRNAs are found to be undesirable for $\beta > 0.0$.

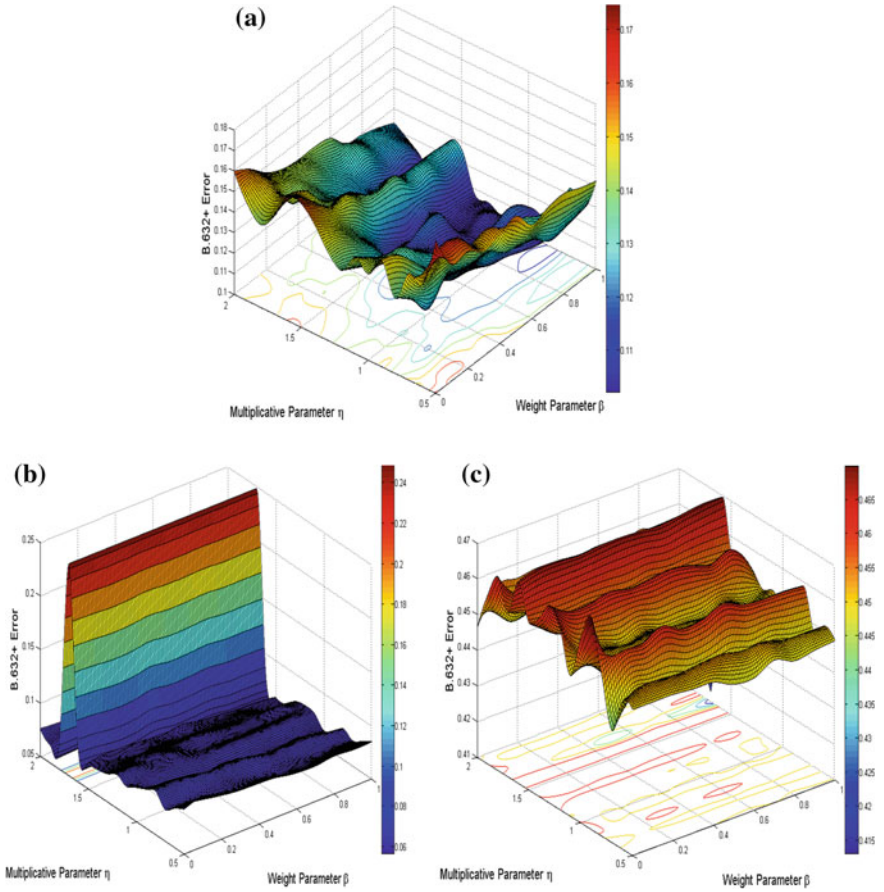

Fig. 7.2 Variation of $B.632+$ error rate of the SVM with respect to multiplicative parameter η and weight parameter β. **a** GSE17681, **b** GSE17846, **c** GSE29352

7.3.3 Importance of B.632+ Error Rate

This section establishes the importance of using $B.632+$ error rate over other types of errors such as apparent error (AE), no-information error rate (γ), and bootstrap error ($B1$). Different types of errors on each miRNA expression data set are calculated using the SVM for the RSMRMS method. Figure 7.3 represents the various types of errors obtained by the RSMRMS algorithm on three miRNA expression data sets. From Fig. 7.3, it is seen that different types of errors decrease as the number of selected miRNAs increases.

For all three miRNA data sets, the AE attains consistently lowest value, while γ has highest value. On the other hand, the $B1$ has smaller error rate than γ but it is

Table 7.1 Optimum values of two parameters for three miRNA data sets

Parameter / Data Set	GSE17681	GSE17846	GSE29352
Weight parameter β	1.0	0.5	1.0
Multiplicative parameter η	1.7	1.0	1.7

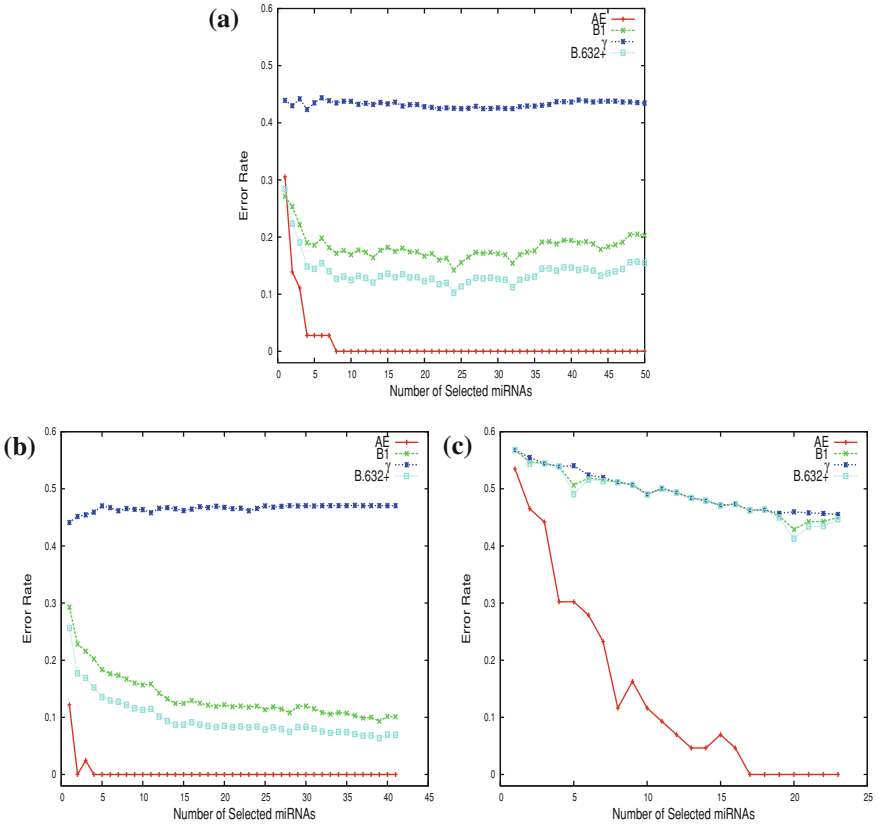

Fig. 7.3 Error rate of the SVM obtained using the RSMRMS algorithm averaged over 50 random splits. **a** GSE17681, **b** GSE17846, **c** GSE29352

higher than the AE. Moreover, the $B.632+$ estimate has smaller error rate than the $B1$ but higher than the AE.

Table 7.2 reports the minimum values of different errors, along with the number of required miRNAs to attain these values. From all the results reported in this table, it can be seen that the $B.632+$ estimator corrects the upward bias of $B1$ and downward bias of AE. Also, it puts more weight on $B1$ in situation where the amount of overfitting as measured by $(B1 - AE)$ is relatively large. It is thus applicable in the present context where the prediction rule generated by the SVM is overfitted.

Table 7.2 Comparative performance analysis of different errors

Different errors	Error/No. of miRNA	Microarray data sets		
		GSE17681	GSE17846	GSE29352
AE	Error	0.000	0.000	0.000
	miRNA	8	2	17
$B1$	Error	0.142	0.093	0.429
	miRNA	24	39	20
γ	Error	0.423	0.441	0.455
	miRNA	4	1	23
$B.632+$	Error	0.103	0.064	0.413
	miRNA	24	39	20

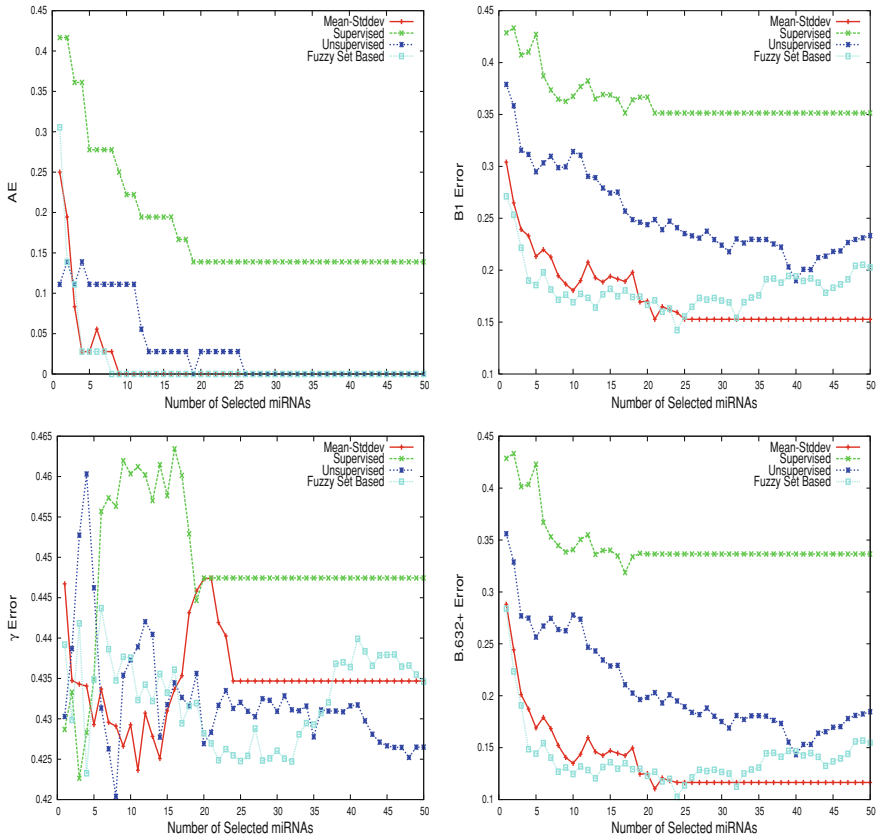

Fig. 7.4 Error rates of the SVM obtained using different discretization methods averaged over 50 random splits for GSE17681

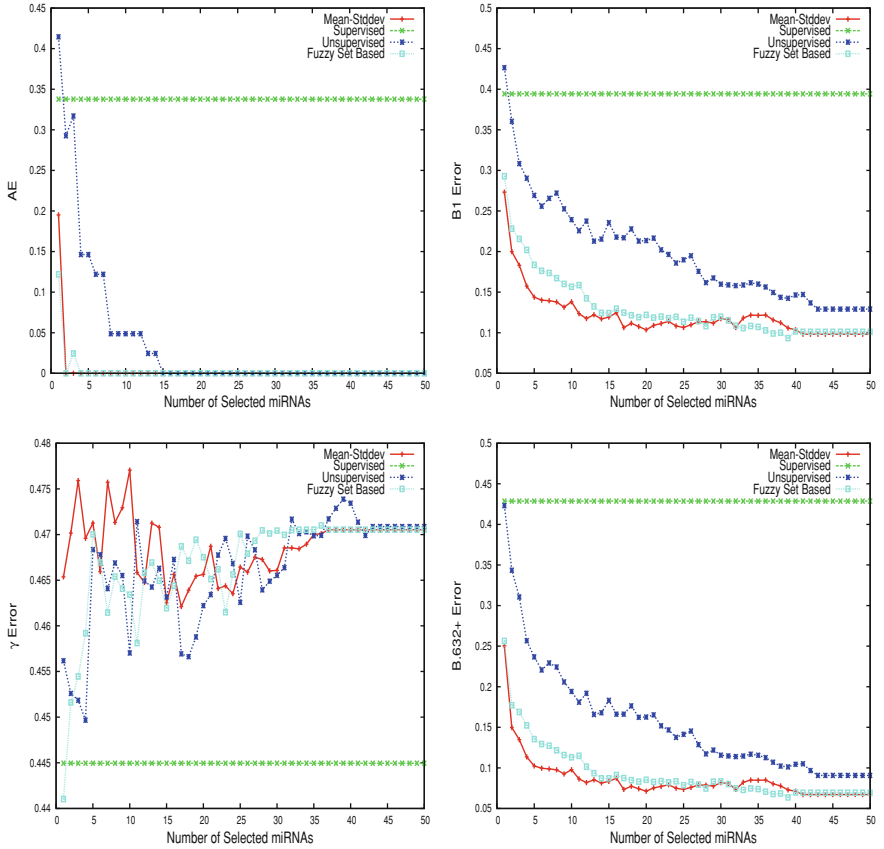

Fig. 7.5 Error rates of the SVM obtained using different discretization methods averaged over 50 random splits for GSE17846

7.3.4 Role of Fuzzy Discretization Method

In the current study, the fuzzy set-based discretization method is used to generate equivalence classes or information granules for computing relevance and significance of miRNAs using the theory of rough sets. To establish the effectiveness of fuzzy set-based discretization method over other discretization methods, extensive experimentation is done on three miRNA data sets. The methods compared are mean and standard deviation-based method (Mean-Stddev) [17] reported in Chaps. 4 and 5, supervised discretization method (Supervised) [22], and unsupervised discretization method (Unsupervised) [22]. Figures 7.4, 7.5, and 7.6 report the variation of several errors with respect to number of selected miRNAs, while Table 7.3 presents the minimum error values obtained using different discretization methods. From all the results reported in Figs. 7.4, 7.5, 7.6 and Table 7.3, it can be seen that the fuzzy

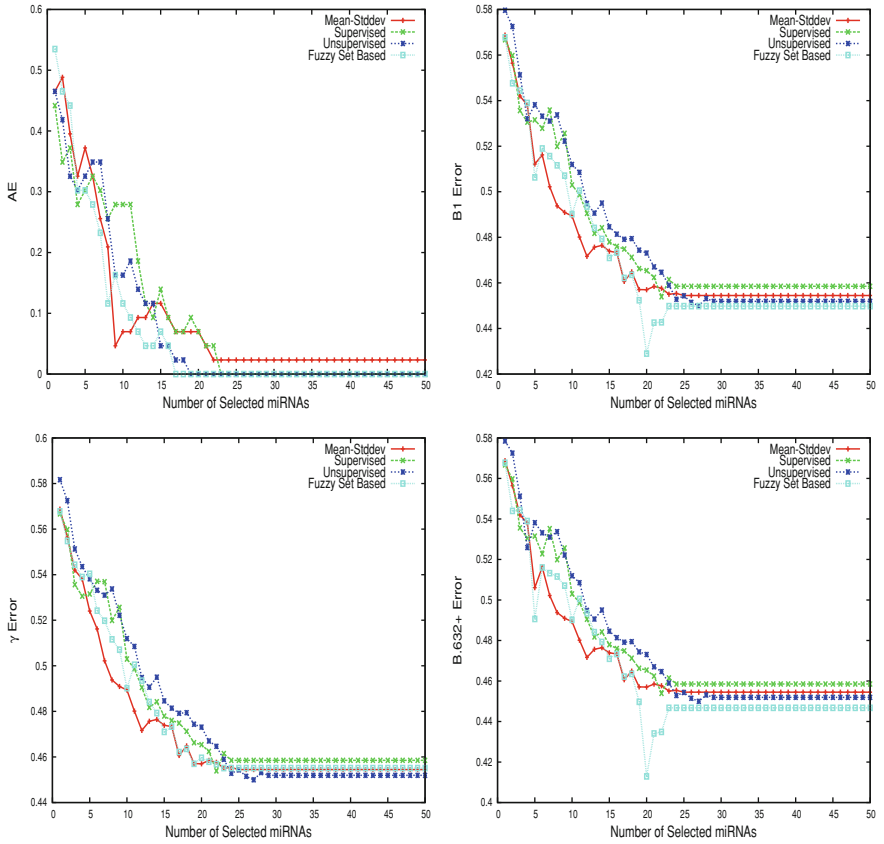

Fig. 7.6 Error rates of the SVM obtained using different discretization methods averaged over 50 random splits for GSE29352

set-based discretion method performs better than other discretization methods, irrespective of the types of errors and miRNA data sets used. However, in case of γ error, the unsupervised discretization method for GSE17681 and GSE29352 data sets and supervised discretization method for GSE29352 data set perform better than the fuzzy set-based discretization method.

7.3.5 Comparative Performance Analysis

This section compares the performance of mRMR and RSMRMS algorithms with respect to various types of errors. Figures 7.7, 7.8, and 7.9 present different error rates obtained by the mRMR and RSMRMS algorithms on three miRNA expression data sets. From all the results reported in Fig. 7.7, 7.8, and 7.9, it is seen that in most

Table 7.3 Comparative performance analysis of different discretization methods

Microarray data sets	Discretization methods	AE		$B1$		γ		$B.632+$	
		Error	miRNAs	Error	miRNAs	Error	miRNAs	Error	miRNAs
GSE17681	Mean-Stddev	0.000	9	0.153	21	0.424	11	0.110	21
	Supervised	0.139	20	0.351	17	0.423	3	0.319	17
	Unsupervised	0.000	26	0.190	40	0.420	8	0.143	40
	Fuzzy set based	0.000	8	0.142	24	0.423	4	0.103	24
GSE17846	Mean-Stddev	0.000	2	0.098	41	0.462	17	0.067	41
	Supervised	0.338	1	0.394	1	0.445	1	0.429	1
	Unsupervised	0.000	15	0.129	43	0.450	4	0.091	43
	Fuzzy set based	0.000	2	0.093	39	0.441	1	0.064	39
GSE29352	Mean-Stddev	0.023	25	0.454	25	0.455	25	0.454	25
	Supervised	0.000	23	0.454	22	0.454	22	0.454	22
	Unsupervised	0.000	19	0.450	27	0.450	27	0.450	27
	Fuzzy set based	0.000	17	0.429	20	0.455	23	0.413	20

Table 7.4 Comparative performance analysis of mRMR and RSMRMS algorithms

Different errors	Error/No. of miRNAs	GSE17681		GSE17846		GSE29352	
		mRMR	RSMRMS	mRMR	RSMRMS	mRMR	RSMRMS
AE	Error	0.00	0.00	0.00	0.00	0.00	0.00
	miRNAs	10	8	3	2	21	17
$B1$	Error	0.18	0.14	0.10	0.09	0.43	0.43
	miRNAs	28	24	48	39	43	20
γ	Error	0.41	0.42	0.44	0.46	0.45	0.46
	miRNAs	13	4	1	5	32	23
$B.632+$	Error	0.13	0.10	0.07	0.06	0.42	0.41
	miRNAs	28	24	48	39	43	20

of the cases different types of error rates are consistently lower for the RSMRMS algorithm compared to the mRMR method.

Finally, Table 7.4 compares the performance of the RSMRMS method with the best performance of the mRMR method. The results are presented based on the error rate of the SVM classifier obtained on three miRNA microarray data sets. From the results reported in Table 7.4, it is seen that although the best AE for each miRNA data set is same for both algorithms, the RSMRMS achieves this value with lower number of selected miRNAs than that obtained by the mRMR method. Also, the RSMRMS attains lowest $B.632+$ bootstrap error rate, as well as $B1$ error rate, of the SVM classifier for all three miRNA data sets with lesser number of selected miRNAs.

The better performance of the RSMRMS algorithm is achieved due to the fact that it uses rough sets for computing both miRNA-class relevance and miRNA-miRNA significance to select differentially expressed miRNAs. The lower and upper approximations of rough sets can effectively deal with incompleteness, vagueness, and

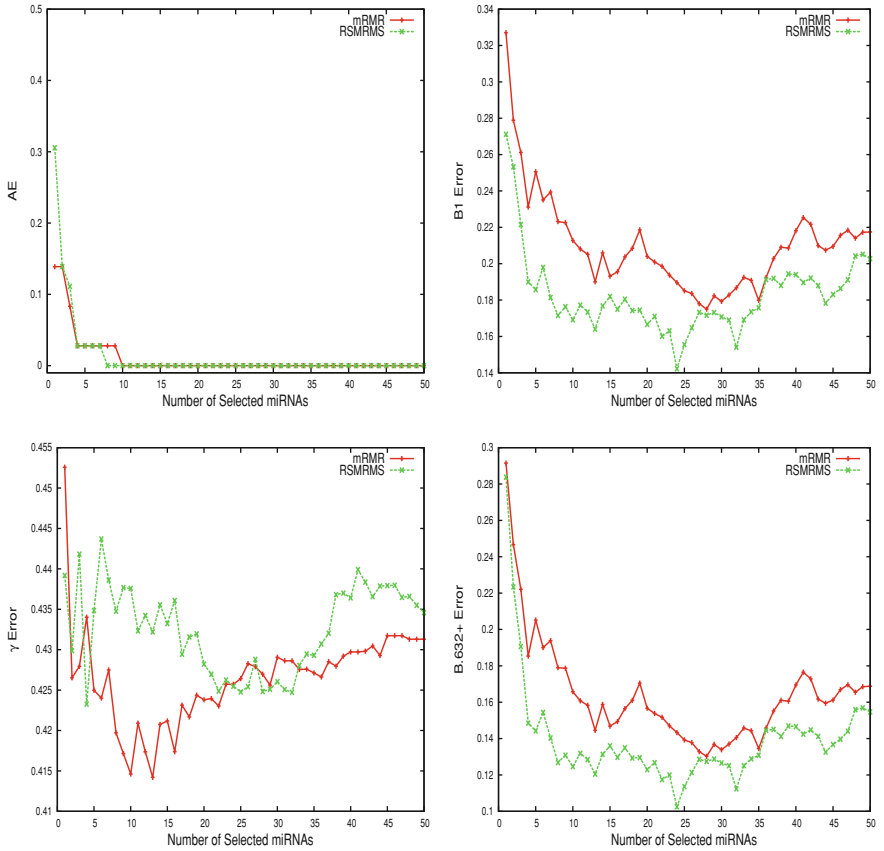

Fig. 7.7 Error rates of the SVM obtained using the mRMR and RSMRMS algorithms averaged over 50 random splits for GSE17681

uncertainty of the data set. The fuzzy set-based discretization method can efficiently handle the overlapping equivalence classes. Also, the mRMR algorithm selects a subset of miRNAs from the whole miRNA set by maximizing the relevance and minimizing the redundancy of the selected miRNAs. The redundancy measure of the mRMR method does not take into account the supervised information of class labels, while both relevance and significance criteria of the RSMRMS method are computed based on the class labels. In effect, the RSMRMS method provides better performance than the mRMR method.

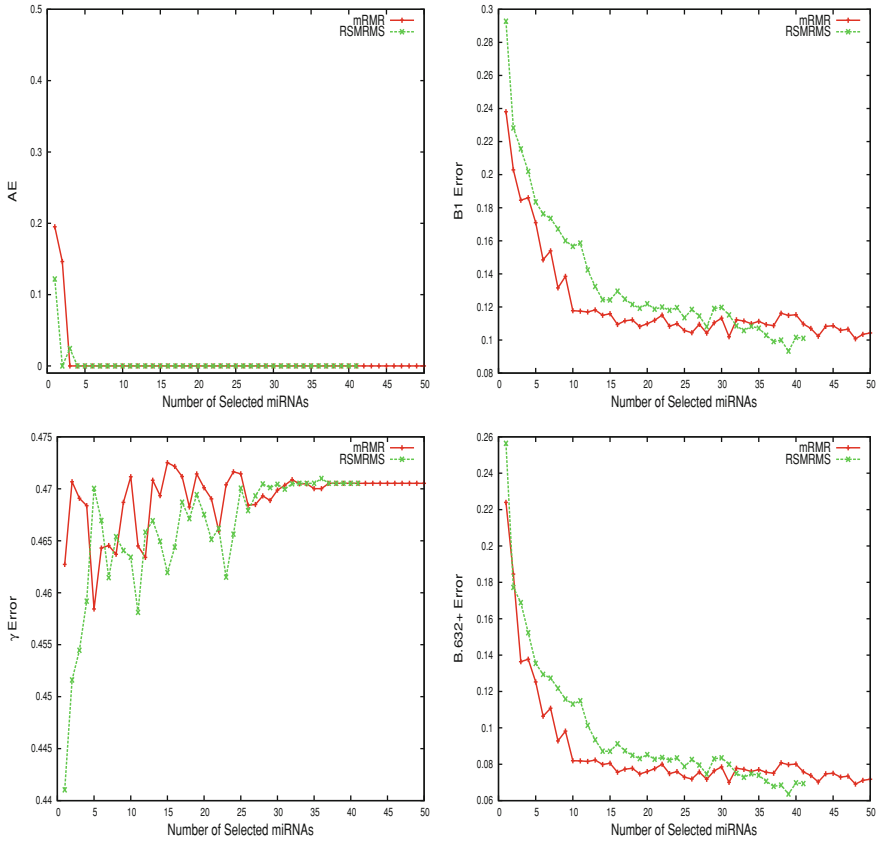

Fig. 7.8 Error rates of the SVM obtained using the mRMR and RSMRMS algorithms averaged over 50 random splits for GSE17846

7.4 Conclusion and Discussion

This chapter addresses the problem of insilico identification of differentially expressed miRNAs. Several existing approaches have been reported along with their merits and demerits. A novel approach is then presented, integrating judiciously the merits of rough set-based maximum relevance-maximum significance (RSMRMS) algorithm, support vector machine, and the $B.632+$ error rate. It selects relevant and significant miRNAs, which can classify samples into different classes with minimum error rate.

All the results reported in this chapter demonstrate the feasibility and effectiveness of the RSMRMS method. The results obtained on three miRNA data sets demonstrate that the RSMRMS method can bring a remarkable improvement on miRNA selection problem, and therefore, it can be a promising alternative to existing models for

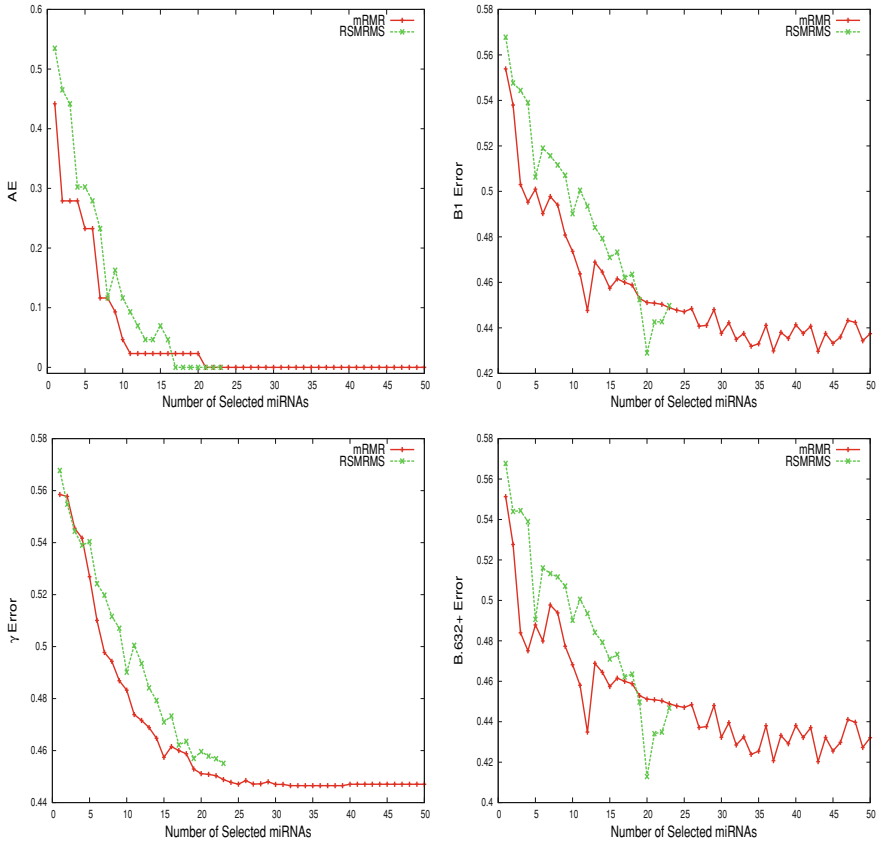

Fig. 7.9 Error rates of the SVM obtained using the mRMR and RSMRMS algorithms averaged over 50 random splits for GSE29352

prediction of class labels of samples. The method is capable of identifying effective miRNAs that may contribute to revealing underlying etiology of a disease, providing a useful tool for exploratory analysis of miRNA data. Recently, Paul and Maji introduced an algorithm using rough hypercuboid equivalence partition matrix for identification of differentially expressed miRNAs [33].

While in Chaps. 4, 5, 6, and 7, we have discussed different feature selection methodologies with extensive experimental results demonstrating their effectiveness in several problems of computational biology and bioinformatics, the next four chapters deal with different clustering approaches, along with some important problems of bioinformatics and medical imaging, namely, grouping functionally similar genes from microarray data, supervised gene clustering for microarray sample classification, possibilistic biclustering for discovering value-coherent overlapping biclusters of genes, and segmentation of brain magnetic resonance images.

References

1. Ambroise C, McLachlan GJ (2002) Selection bias in gene extraction on the basis of microarray gene-expression data. Proc Nat Acad Sci USA 99(10):6562–6566
2. Arora S, Ranade AR, Tran NL, Nasser S, Sridhar S, Korn RL, Ross JTD, Dhruv H, Foss KM, Sibenaller Z, Ryken T, Gotway MB, Kim S, Weiss GJ (2011) MicroRNA-328 is associated with non-small cell lung cancer (NSCLC) brain metastasis and mediates NSCLC migration. Int J Cancer 129(11):2621–2631
3. Blenkiron C, Goldstein LD, Thorne NP, Spiteri I, Chin SF, Dunning MJ, Barbosa-Morais NL, Teschendorff AE, Green AR, Ellis IO, Tavare S, Caldas C, Miska EA (2007) MicroRNA expression profiling of human breast cancer identifies new markers of tumor subtype. Genome Biol 8(10):214.1–214.16
4. Budhu A, Ji J, Wang XW (2010) The clinical potential of microRNAs. J Hematol Oncol 3(37):1–7
5. Chen Y, Stallings RL (2007) Differential patterns of microRNA expression in neuroblastoma are correlated with prognosis, differentiation, and apoptosis. Cancer Res 67:976–983
6. Ding C, Peng H (2005) Minimum redundancy feature selection from microarray gene expression data. J Bioinform Comput Biol 3(2):185–205
7. Efron B, Tibshirani R (1997) Improvements on cross-validation: the. 632+ bootstrap method. J Am Stat Assoc 92(438):548–560
8. Fang J, Busse JWG (2006) Mining of microRNA expression data - a rough set approach. In: Proceedings of the 1st international conference on rough sets and knowledge technology, Springer, Berlin, pp 758–765
9. Guo J, Miao Y, Xiao B, Huan R, Jiang Z, Meng D, Wang Y (2009) Differential expression of microRNA species in human gastric cancer versus non-tumorous tissues. J Gastroenterol Hepatol 24:652–657
10. Iorio MV, Visone R, Leva GD, Donati V, Petrocca F, Casalini P, Taccioli C, Volinia S, Liu CG, Alder H, Calin GA, Menard S, Croce CM (2007) MicroRNA signatures in human ovarian cancer. Cancer Res 67(18):8699–8707
11. Keller A, Leidinger P, Lange J, Borries A, Schroers H, Scheffler M, Lenhof HP, Ruprecht K, Meese E (2009) Multiple sclerosis: microRNA expression profiles accurately differentiate patients with relapsing-remitting disease from healthy controls. PLoS ONE 4(10):e7440
12. Keller A, Leidinger P, Wendschlag A, Scheffler M, Meese E, Wucherpfennig F, Huwer H, Borries A (2009) miRNAs in lung cancer—studying complex fingerprints in patient's blood cells by microarray experiments. BMC Cancer 9:353
13. Klir G, Yuan B (2005) Fuzzy sets and fuzzy logic: theory and applications. Prentice Hall, New Delhi, India
14. Lehmann U, Streichert T, Otto B, Albat C, Hasemeier B, Christgen H, Schipper E, Hille U, Kreipe HH, Langer F (2010) Identification of differentially expressed microRNAs in human male breast cancer. BMC Bioinformatics 10:1–9
15. Li S, Chen X, Zhang H, Liang X, Xiang Y, Yu C, Zen K, Li Y, Zhang CY (2009) Differential expression of microRNAs in mouse liver under aberrant energy metabolic status. J Lipid Res 50:1756–1765
16. Lu J, Getz G, Miska EA, Saavedra EA, Lamb J, Peck D, Cordero AS, Ebert BL, Mak RH, Ferrando AA, Downing JR, Jacks T, Horvitz HR, Golub TR (2005) MicroRNA expression profiles classify human cancers. Nat Lett 435(9):834–838
17. Maji P (2009) f-Information measures for efficient selection of discriminative genes from microarray data. IEEE Trans Biomed Eng 56(4):1063–1069
18. Maji P (2011) Fuzzy-rough supervised attribute clustering algorithm and classification of microarray data. IEEE Trans Syst Man Cybern Part B Cybern 41(1):222–233
19. Maji P, Das C (2012) Relevant and significant supervised gene clusters for microarray cancer classification. IEEE Trans NanoBiosci 11(2):161–168
20. Maji P, Pal SK (2010) Feature selection using f-information measures in fuzzy approximation spaces. IEEE Trans Knowl Data Eng 22(6):854–867

21. Maji P, Pal SK (2010) Fuzzy-rough sets for information measures and selection of relevant genes from microarray data. IEEE Trans Syst Man Cybern Part B Cybern 40(3):741–752
22. Maji P, Pal SK (2012) Rough-fuzzy pattern recognition: applications in bioinformatics and medical imaging. Wiley-IEEE Computer Society Press, New Jersey
23. Maji P, Paul S (2011) Microarray time-series data clustering using rough-fuzzy c-means algorithm. In: Proceedings of the 5th IEEE international conference on bioinformatics and biomedicine, Atlanta, pp 269–272
24. Maji P, Paul S (2011) Rough set based maximum relevance-maximum significance criterion and gene selection from microarray data. Int J Approximate Reasoning 52(3):408–426
25. Maji P, Paul S (2013) Rough-fuzzy clustering for grouping functionally similar genes from microarray data. IEEE/ACM Trans Comput Biol Bioinf 10(2):286–299
26. McIver AD, East P, Mein CA, Cazier JB, Molloy G, Chaplin T, Lister TA, Young BD, Debernardi S (2008) Distinctive patterns of microRNA expression associated with karyotype in acute myeloid leukaemia. PLoS ONE 3(5):1–8
27. Nasser S, Ranade AR, Sridhart S, Haney L, Korn RL, Gotway MB, Weiss GJ, Kim S (2009) Identifying miRNA and imaging features associated with metastasis of lung cancer to the brain. In: Proceedings of IEEE international conference on bioinformatics and biomedicine, pp 246–251
28. Ortega FJ, Moreno-Navarrete JM, Pardo G, Sabater M, Hummel M, Ferrer A, Rodriguez-Hermosa JI, Ruiz B, Ricart W, Peral B, Real JMF (2010) MiRNA expression profile of human subcutaneous adipose and during adipocyte differentiation. PLoS ONE 5(2):1–9
29. Pal SK, Mitra S (1999) Neuro-fuzzy pattern recognition: methods in soft computing. Wiley, New York
30. Pal SK, Pramanik PK (1986) Fuzzy measures in determining seed points in clustering. Pattern Recogn Lett 4(3):159–164
31. Paul S, Maji P (2012) Robust RFCM algorithm for identification of co-expressed miRNAs. In: Proceedings of IEEE international conference on bioinformatics and biomedicine, Philadelphia, pp 520–523
32. Paul S, Maji P (2012) Rough sets and support vector machine for selecting differentially expressed miRNAs. In: Proceedings of IEEE international conference on bioinformatics and biomedicine workshops: nanoinformatics for biomedicine, Philadelphia, pp 864–871
33. Paul S, Maji P (2013) μHEM for identification of differentially expressed miRNAs using hypercuboid equivalence partition matrix. BMC Bioinformatics 14(1):266
34. Paul S, Maji P (2013) Rough sets for insilico identification of differentially expressed miRNAs. Int J Nanomed 8:63–74
35. Pawlak Z (1991) Rough sets: theoretical aspects of resoning about data. Kluwer, Dordrecht
36. Pereira PM, Marques JP, Soares AR, Carreto L, Santos MAS (2010) MicroRNA expression variability in human cervical tissues. PLoS ONE 5(7):1–12
37. Raponi M, Dossey L, Jatkoe T, Wu X, Chen G, Fan H, Beer DG (2009) MicroRNA classifiers for predicting prognosis of squamous cell lung cancer. Cancer Res 69(14):5776–5783
38. Schrauder MG, Strick R, Schulz-Wendtland R, Strissel PL, Kahmann L, Loehberg CR, Lux MP, Jud SM, Hartmann A, Hein A, Bayer CM, Bani MR, Richter S, Adamietz BR, Wenkel E, Rauh C, Beckmann MW, Fasching PA (2012) Circulating microRNAs as potential blood-based markers for early stage breast cancer detection. PLoS ONE 7(1):1–9
39. Slezak D, Wroblewski J (2007) Roughfication of numeric decision tables: the case study of gene expression data. In: Proceedings of the 2nd international conference on rough sets and knowledge technology, Springer, Berlin, pp 316–323
40. Valdes JJ, Barton AJ (2006) Relevant attribute discovery in high dimensional data: application to breast cancer gene expressions. In: Proceedings of the 1st international conference on rough sets and knowledge technology, Springer, Berlin, pp 482–489
41. Vapnik V (1995) The nature of statistical learning theory. Springer, New York
42. Wang C, Yang S, Sun G, Tang X, Lu S, Neyrolles O, Gao Q (2011) Comparative miRNA expression profiles in individuals with latent and active tuberculosis. PLoS ONE 6(10):1–11

43. Xu R, Xu J, Wunsch DC (2009) MicroRNA expression profile based cancer classification using default ARTMAP. Neural Netw 22:774–780
44. Zadeh LA (1965) Fuzzy sets. Inf Control 8:338–353
45. Zhao H, Shen J, Medico L, Wang D, Ambrosone CB, Liu S (2010) A pilot study of circulating miRNAs as potential biomarkers of early stage breast cancer. PLoS ONE 5(10):1–12
46. Zhu M, Yi M, Kim CH, Deng C, Li Y, Medina D, Stephens RM, Green JE (2011) Integrated miRNA and mRNA expression profiling of mouse mammary tumor models identifies miRNA signatures associated with mammary tumor lineage. Genome Biol 12:1–16

Part III
Clustering

Chapter 8
Grouping Functionally Similar Genes From Microarray Data Using Rough–Fuzzy Clustering

8.1 Introduction

Microarray technology is one of the important biotechnological means that has made it possible to simultaneously monitor the expression levels of thousands of genes during important biological processes and across collections of related samples [9, 14, 37]. An important application of microarray data is to elucidate the patterns hidden in gene expression data for an enhanced understanding of functional genomics.

A microarray time-series gene expression data set can be represented by an expression table, where each row corresponds to one particular gene, each column to a time point, and each entry of the matrix is the measured expression level of a particular gene in a time point, respectively [9, 14, 37]. However, the large number of genes and the complexity of biological networks greatly increase the challenges of comprehending and interpreting the resulting mass of data, which often consists of millions of measurements. A first step toward addressing this challenge is the use of clustering techniques, which is essential in pattern recognition process to reveal natural structures and identify interesting patterns in the underlying data [28].

Cluster analysis is a technique for finding natural groups present in the data. It divides a given data set into a set of clusters in such a way that two objects from the same cluster are as similar as possible and the objects from different clusters are as dissimilar as possible. In effect, it tries to mimic the human ability to group similar objects into classes and categories [25]. Clustering techniques have been effectively applied to a wide range of engineering and scientific disciplines such as pattern recognition, machine learning, psychology, biology, medicine, computer vision, web intelligence, communications, and remote sensing. A number of clustering algorithms have been proposed to suit different requirements [25, 26].

The purpose of gene clustering is to group together coexpressed genes which indicate cofunction and coregulation. Due to the special characteristics of gene expression data, and the particular requirements from the biological domain, gene clustering presents several new challenges and is still an open problem. To understand gene

P. Maji and S. Paul, *Scalable Pattern Recognition Algorithms*,
DOI: 10.1007/978-3-319-05630-2_8,

function, gene regulation, cellular processes, and subtypes of cells, clustering techniques have proven to be helpful. The coexpressed genes, that is, genes with similar expression patterns, can be clustered together with similar cellular functions. This approach may further help to understand the functions of many genes for which information has not been previously available [17, 54]. Furthermore, coexpressed genes in the same cluster are likely to be involved in the same cellular processes, and a strong correlation of expression patterns between those genes indicates coregulation. Searching for common DNA sequences at the promoter regions of genes within the same cluster, allows regulatory motifs specific to each gene cluster to be identified and cis-regulatory elements to be proposed [8, 54]. The inference of regulation through gene expression data clustering also gives rise to hypotheses regarding the mechanism of transcriptional regulatory network [13].

Different clustering techniques such as hierarchical clustering [22], hard c-means or k-means algorithm [23], self organizing map [53], principal component analysis [42], graph theoretical approaches [5, 21, 51, 58], model-based clustering [18, 20, 41, 59], and density-based approach [27] have been widely applied to find groups of coexpressed genes from microarray data. A comprehensive survey on various gene clustering algorithms can be found in [8, 28]. One of the most widely used prototype-based partitional clustering algorithms is hard c-means [43]. In hard c-means, each gene is assigned to exactly one cluster. However, one of the main problems in gene expression data analysis is uncertainty. Some of the sources of this uncertainty include incompleteness and vagueness in cluster definitions. Also, the empirical study has demonstrated that gene expression data are often highly connected, and the clusters may be highly overlapping with each other or even embedded one in another [27]. Therefore, gene clustering algorithms should be able to effectively handle this situation. Moreover, gene expression data often contains a huge amount of noise due to the complex procedures of microarray experiments [29]. Hence, clustering algorithms for gene expression data should be capable of extracting useful information from a high level of background noise.

In this background, the possibility concept introduced by fuzzy set theory [60] and rough set theory [46] have gained popularity in modeling and propagating uncertainty. Both fuzzy sets and rough sets provide a mathematical framework to capture uncertainties associated with the data. One of the most notable prototype-based partitional clustering algorithms is fuzzy c-means [6, 16]. It assigns each gene to every cluster by allowing gradual memberships. In effect, it offers the opportunity to deal with the data that belong to more than one cluster at the same time. It assigns memberships to a gene which are inversely related to the relative distance of the gene to cluster prototypes. Also, it can deal with the uncertainties arising from overlapping cluster boundaries and reveal additional information concerning gene coexpression [12, 15, 19, 57]. In particular, information regarding overlapping clusters and overlapping cellular pathways has been identified from fuzzy clustering results [4, 19]. However, the resulting membership values of fuzzy c-means do not always correspond well to the degrees of belonging of the data, and it may be inaccurate in a noisy environment [30]. To reduce this weakness and to produce memberships that have a good explanation of the degrees of belonging for the data, Krishnapuram

and Keller [30] proposed possibilistic c-means algorithm. However, it sometimes generates coincident clusters [37].

On the other hand, two of the early rough clustering algorithms are those due to Hirano and Tsumoto [24] and De [11]. Other notable algorithms include rough c-means [33], rough self organizing map [44], and rough support vector clustering [2]. In [45], the indiscernibility relation of rough sets has been used to initialize the expectation maximization algorithm. In [33], Lingras and West introduced a rough clustering method, called rough c-means, which describes a cluster by a prototype or center and a pair of lower and upper approximations. The lower and upper approximations are weighted different parameters to compute the new centers. Asharaf et al. [1] extended rough c-means algorithm that may not require specification of the number of clusters.

Integrating the merits of rough sets and fuzzy sets, different rough–fuzzy clustering algorithms such as rough–fuzzy c-means [35], rough-possibilistic c-means [36], and rough–fuzzy-possibilistic c-means [36] have been proposed, where each cluster is represented by a cluster prototype, a crisp lower approximation and a probabilistic and/or possibilistic fuzzy boundary. The cluster prototype is computed based on the weighted average of crisp lower approximation and fuzzy boundary. All these algorithms can be used for clustering coexpressed genes from microarray gene expression data sets [37, 38]. Recently, a robust rough–fuzzy clustering algorithm is proposed in [40] to group functionally similar genes. Also, fuzzy–rough supervised gene clustering algorithm is proposed in [34] to find groups of coregulated genes whose collective expression is strongly associated with sample categories.

This chapter presents the application of rough–fuzzy c-means for clustering functionally similar genes from microarray time-series gene expression data sets. An efficient method, proposed by Maji and Paul [39], is reported to select initial prototypes of different gene clusters; thereby circumventing the initialization and local minima problems of different c-means algorithms. A parameter optimization method is also presented based on cluster validity index to identify optimum values of different parameters of the initialization method and the clustering algorithms. The effectiveness of different partitive clustering algorithms is demonstrated on a set of five yeast microarray gene expression data sets using some cluster validity indices and gene ontology-based analysis.

The rest of this chapter is organized as follows: Sect. 8.2 presents a survey on different gene clustering algorithms and several quantitative measures for evaluating the clustering solutions. Section 8.3 presents rough–fuzzy c-means algorithm, along with a new initialization method for selection of initial cluster prototypes of different partitive clustering algorithms and a parameter optimization technique based on cluster validity index. Experimental results, a brief description of different yeast microarray gene expression data sets, and a comparison among several gene clustering algorithms are presented in Sect. 8.4. The biological importance of each clustering solution is evaluated using gene ontology. Concluding remarks are given in Sect. 8.5.

8.2 Clustering Algorithms and Validity Indices

Clustering is one of the major tasks in gene expression data analysis. When applied to gene expression data, clustering algorithms can be applied on both gene and sample dimensions [3, 28]. The conventional clustering methods group a subset of genes that are interdependent or correlated with each other. In other words, genes in a cluster are more correlated with each other, whereas genes in different clusters are less correlated [3]. After clustering genes, a reduced set of genes can be selected for further analysis. The conventional gene clustering methods allow genes with similar expression patterns, that is, coexpressed genes, to be identified [28].

8.2.1 Different Gene Clustering Algorithms

This section presents a brief overview on different clustering algorithms, which can be used to group coexpressed genes from microarray expression data sets.

8.2.1.1 Hard C-Means

The hard c-means [43] is one of the simplest unsupervised learning algorithms. Let $X = \{x_1, \ldots, x_j, \ldots, x_n\}$ and $V = \{v_1, \ldots, v_i, \ldots, v_c\}$ be the set of n objects and c centroids, respectively, having m dimensions where $x_j \in \Re^m$ and $v_i \in \Re^m$. The objective of hard c-means algorithm is to assign n objects to c clusters. Each of the clusters β_i is represented by a centroid v_i, which is the cluster representative for that cluster. The process begins by randomly choosing c objects as the centroids or means. The objects are assigned to one of the c clusters based on the similarity or dissimilarity between the object x_j and the centroid v_i. After the assignment of all the objects to various clusters, the new centroids are calculated as follows:

$$v_i = \frac{1}{n_i} \sum_{x_j \in \beta_i} x_j, \tag{8.1}$$

where n_i represents the number of objects in cluster β_i. The main steps of hard c-means algorithm are as follows:

1. Assign initial means or centroids $v_i, i = 1, 2, \ldots, c$.
2. For each object x_j, calculate distance d_{ij} between itself and the centroid v_i of cluster β_i.
3. If d_{ij} is minimum for $1 \leq i \leq c$, then $x_j \in \beta_i$.
4. Compute new centroid as per (8.1).
5. Repeat steps 2 to 4 until no more new assignments can be made.

8.2.1.2 Fuzzy C-Means

The fuzzy c-means provides a fuzzification of the hard c-means [6]. It partitions X into c clusters by minimizing the following objective function

$$J_F = \sum_{j=1}^{n}\sum_{i=1}^{c}(\mu_{ij})^{\acute{m}_1}||x_j - v_i||^2 \tag{8.2}$$

where $\acute{m}_1 \in [1, \infty)$ is the fuzzifier, v_i is the ith centroid corresponding to cluster β_i, $\mu_{ij} \in [0, 1]$ is the probabilistic membership of the pattern x_j to cluster β_i, and $||.||$ is the distance norm such that

$$v_i = \frac{1}{n_i}\sum_{j=1}^{n}(\mu_{ij})^{\acute{m}_1}x_j; \tag{8.3}$$

where

$$n_i = \sum_{j=1}^{n}(\mu_{ij})^{\acute{m}_1} \tag{8.4}$$

and

$$\mu_{ij} = \left(\sum_{k=1}^{c}\left(\frac{d_{ij}^2}{d_{kj}^2}\right)^{\frac{1}{\acute{m}_1-1}}\right)^{-1}; \tag{8.5}$$

where

$$d_{ij}^2 = ||x_j - v_i||^2 \tag{8.6}$$

subject to

$$\sum_{i=1}^{c}\mu_{ij} = 1, \forall j, \text{ and } 0 < \sum_{j=1}^{n}\mu_{ij} < n, \forall i. \tag{8.7}$$

The process begins by randomly choosing c objects as the centroids or means of the c clusters. The memberships are calculated based on the relative distance of the object x_j to the centroids $\{v_i\}$ by (8.5). After computing memberships of all the objects, the new centroids of the clusters are calculated as per (8.3). The process stops when the centroids stabilize. That is, the centroids from the previous iteration are identical to those generated in the current iteration. The basic steps are outlined as follows:

1. Assign initial means v_i, $i = 1, 2, \ldots, c$. Choose values for $\acute{m}_1$ and threshold ε. Set iteration counter $t = 1$.
2. Compute memberships μ_{ij} by (8.5) for c clusters and n objects.
3. Update mean or centroid v_i by (8.3).
4. Repeat steps 2 and 3, by incrementing t, until $|\mu_{ij}(t) - \mu_{ij}(t-1)| > \varepsilon$.

8.2.1.3 Possibilistic C-Means

In fuzzy c-means, the memberships of an object are inversely related to the relative distance of the object to the cluster centroids. In effect, it is very sensitive to noise and outliers. Also, from the standpoint of compatibility with the centroid, the memberships of an object x_j in a cluster β_i should be determined solely by how close it is to the mean or centroid v_i of the class, and should not be coupled with its similarity with respect to other classes.

To alleviate this problem, Krishnapuram and Keller [30, 31] introduced possibilistic c-means, where the objective function can be formulated as

$$J_P = \sum_{i=1}^{c}\sum_{j=1}^{n}(v_{ij})^{\acute{m}_2}||x_j - v_i||^2 + \sum_{i=1}^{c}\eta_i\sum_{j=1}^{n}(1 - v_{ij})^{\acute{m}_2} \tag{8.8}$$

where $\acute{m}_2 \in [1, \infty)$ is the fuzzifier and η_i represents the scale parameter. The membership matrix v generated by possibilistic c-means is not a partition matrix in the sense that it does not satisfy the constraint

$$\sum_{i=1}^{c} v_{ij} = 1, \forall j. \tag{8.9}$$

The update equation of v_{ij} is given by

$$v_{ij} = \frac{1}{1 + \mathrm{D}}; \tag{8.10}$$

where

$$\mathrm{D} = \left\{\frac{||x_j - v_i||^2}{\eta_i}\right\}^{\frac{1}{\acute{m}_2 - 1}} \tag{8.11}$$

subject to

$$v_{ij} \in [0, 1], \forall i, j; 0 < \sum_{j=1}^{n} v_{ij} \leq n, \forall i; \text{ and } \max_i v_{ij} > 0, \forall j. \tag{8.12}$$

The scale parameter η_i represents the zone of influence or size of the cluster β_i. The update equation for η_i is given by

$$\eta_i = \mathrm{K} \cdot \frac{\mathrm{P}}{\mathrm{Q}} \tag{8.13}$$

where

$$\mathrm{P} = \sum_{j=1}^{n} (v_{ij})^{m'_2} ||x_j - v_i||^2; \tag{8.14}$$

and

$$\mathrm{Q} = \sum_{j=1}^{n} (v_{ij})^{m'_2}. \tag{8.15}$$

Typically K is chosen to be one. In each iteration, the updated value of v_{ij} depends only on the similarity between the object x_j and the centroid v_i. The resulting partition of the data can be interpreted as a possibilistic partition, and the membership values may be interpreted as degrees of possibility of the objects belonging to the classes, that is, the compatibilities of the objects with the means or centroids. The updating of the means proceeds exactly the same way as in the case of fuzzy c-means algorithm.

8.2.1.4 Self Organizing Map

The self organizing map (SOM) creates a set of prototype vectors representing the data set and carries out a topology preserving projection of the prototypes from the d-dimensional input space onto a low-dimensional grid. This ordered grid represents the cluster structures. The main issue associated with the SOM is that it requires a prespecified number and an initial spatial structure of clusters [53]. However, it is difficult to prior estimate these values. Furthermore, if the data set is abundant with irrelevant data points such as genes with invariant patterns, the SOM may produce an output in which this type of data will populate the vast majority of clusters. In this case, the SOM is not effective because most of the interesting patterns may be merged into only one or two clusters and cannot be identified [28].

8.2.1.5 Graph Theoretical Approaches

Graph theoretical clustering techniques are explicitly presented in terms of a graph, where each gene corresponds to a vertex, while for some clustering methods, each pair of genes is connected by an edge with weight assigned according to the proximity value between the genes [51, 58]. For other methods, proximity is mapped only to either zero or one on the basis of some threshold [5, 21]. Hence, this approach

converts the problem of clustering a gene set into graph theoretical problems as finding minimum cut or maximal cliques in the proximity graph.

The CLICK (CLuster Identification via Connectivity Kernels) [51] seeks to identify highly connected components in the proximity graph as clusters. It makes the probabilistic assumption that after standardization, pairwise similarity values between elements are normally distributed, no matter if they are in the same cluster or not. Under this assumption, the weight of an edge between two vertices is defined as the probability that two vertices are in the same cluster. The clustering process of the CLICK iteratively finds the minimum cut in the proximity graph and recursively splits the gene set into a set of connected components from the minimum cut. The CLICK also takes two postpruning steps, namely, adoption and merging, to refine the clustering results. The adoption step handles the remaining singletons and updates the current clusters, while the merging step iteratively merges two clusters with similarity exceeding a predefined threshold.

8.2.1.6 Hierarchical, Model, and Density-Based Approaches

The partition-based clustering such as k-means algorithm directly decomposes the data set into a set of disjoint clusters. In contrast, hierarchical clustering generates a hierarchical series of nested clusters that can be graphically represented by a tree, called dendrogram. The branches of a dendrogram not only record the formation of the clusters but also indicate the similarity between the clusters. By cutting the dendrogram at some level, one can obtain a specified number of clusters. By reordering the objects such that the branches of the corresponding dendrogram do not cross, the data set can be arranged with similar objects placed together [28]. The hierarchical clustering identifies sets of correlated genes with similar behavior across the samples, but yields thousands of clusters in a tree-like structure, which makes the identification of functional groups very difficult [22].

The CAST (Cluster Affinity Search Technique) [5] has been found to be an attractive clustering procedure for cluster detection in gene expression data. The CAST is a sequential procedure that defines clusters one at a time. After detecting a cluster, the CAST removes the corresponding genes from consideration and initializes the next cluster. Cluster detection proceeds by adding and removing genes based on a similarity measure between the genes and the cluster members. A gene is added if its similarity to the cluster exceeds a user-defined threshold, that suggests, that the gene cluster has high affinity for the cluster. At each iteration, the low affinity gene will be dropped if its similarity to the cluster falls below the threshold. Note that the threshold is constant for the whole clustering procedure, thus it is referred to as global affinity threshold. However, the CAST has the usual difficulty of determining a good value for the global affinity threshold [28].

The model-based clustering approaches [18, 20, 41, 59] provide a statistical framework to model the cluster structure of gene expression data. The data set is assumed to come from a finite mixture of underlying probability distributions with each component corresponding to a different cluster. The goal is to estimate the

parameters that maximize the likelihood. Usually, the parameters are estimated by the expectation maximization (EM) algorithm. The EM algorithm iterates between Expectation (E) and Maximization (M) steps. In the E step, hidden parameters are conditionally estimated from the data with the current estimated model parameter. In the M step, model parameters are estimated so as to maximize the likelihood of complete data given the estimated hidden parameters. When the EM algorithm converges, each gene is assigned to the component or cluster with the maximum conditional probability [28].

A density-based hierarchical clustering method, called the DHC, is proposed in [27] to identify the coexpressed gene groups from gene expression data. The DHC is developed based on the notions of density and attraction of data objects. The basic idea is to consider a cluster as a high-dimensional dense area, where objects are attracted with each other. At the core part of the dense area, objects are crowded closely with each other and thus have high density. Objects at the peripheral area of the cluster are relatively sparsely distributed and are attracted to the core part of the dense area. Once the density and attraction of data objects are defined, the DHC organizes the cluster structure of the data set in two-level hierarchical structures [28].

8.2.2 Quantitative Measures

Following quantitative indices are generally used, along with other measures [28], to evaluate the performance of different gene clustering algorithms for grouping functionally similar genes from microarray gene expression data sets.

8.2.2.1 Silhouette Index

Let an object $x_i \in \beta_r$, $i = 1, \ldots, n_r$ and n_r is the cardinality of cluster β_r. For each object x_i let a_i be the average distance between object x_i and rest of the objects of β_r, that is,

$$a_i = d_{\text{avg}}(x_i, \beta_r - \{x_i\}) \tag{8.16}$$

where $d_{\text{avg}}(.,.)$ denotes the average distance measure between an object and a set of objects. For any other cluster $\beta_p \neq \beta_r$, let $d_{\text{avg}}(x_i, \beta_p)$ denote the average distance of object x_i to all objects of β_p. The scalar b_i is the smallest of these $d_{\text{avg}}(x_i, \beta_p)$, $p = 1, \ldots, c$, $p \neq r$, that is,

$$b_i = \min_{p=1,\ldots,c,\, p\neq r} \{d_{\text{avg}}(x_i, \beta_p)\}. \tag{8.17}$$

The Silhouette width of object x_i is then defined as [48]

$$s(x_i) = \frac{b_i - a_i}{\max\{b_i, a_i\}} \tag{8.18}$$

where $-1 \leq s(x_i) \leq 1$. The value of $s(x_i)$ close to 1 implies that the distance of object x_i from the cluster β_r where it belongs is significantly less than the distance between x_i and its nearest cluster excluding β_r, which indicates that x_i is well clustered. On the other hand, the value of $s(x_i)$ close to -1 implies that the distance between x_i and β_r is significantly higher than the distance between x_i and its nearest cluster excluding β_r, which indicates that x_i is not well clustered. Finally, the values of $s(x_i)$ close to 0 indicate that x_i lies close to the border between the two clusters. Based on the definition of $s(x_i)$, the Silhouette of the cluster β_k $(k = 1, \ldots, c)$ is defined as

$$S(\beta_k) = \frac{1}{n_k} \sum_{x_i \in \beta_k} s(x_i) \tag{8.19}$$

where n_k is the cardinality of the cluster β_k. The global Silhouette index is defined as

$$\hat{S}_c = \frac{1}{c} \sum_{k=1}^{c} S(\beta_k) \tag{8.20}$$

where $\hat{S}_c \in [-1, 1]$. Also, the higher the value of $\hat{S}_c$, the better the corresponding clustering is.

8.2.2.2 Davies-Bouldin Index

The Davies-Bouldin (DB) index [10] is a function of the ratio of sum of within-cluster distance to between-cluster separation and is given by

$$\text{DB} = \frac{1}{c} \sum_{i=1}^{c} \max_{k \neq i} \left\{ \frac{S(v_i) + S(v_k)}{d(v_i, v_k)} \right\} \tag{8.21}$$

for $1 \leq i, k \leq c$. The DB index minimizes the within-cluster distance $S(v_i)$ and maximizes the between-cluster separation $d(v_i, v_k)$. Therefore, for a given data set and c value, the higher the similarity values within the clusters and the between-cluster separation, lower would be the DB index value. A good clustering procedure should make the value of DB index as low as possible.

8.2.2.3 Dunn Index

Dunn's index [16] is also designed to identify sets of clusters that are compact and well separated. Dunn's (D) index maximizes

$$\mathrm{D} = \min_{i} \left\{ \min_{k \neq i} \left\{ \frac{d(v_i, v_k)}{\max_l S(v_l)} \right\} \right\} \tag{8.22}$$

for $1 \leq i, k, l \leq c$. A good clustering procedure should make the value of Dunn index as high as possible.

8.3 Grouping Functionally Similar Genes Using Rough–Fuzzy C-Means Algorithm

This section presents a new c-means algorithm based on both fuzzy and rough sets, termed as rough–fuzzy c-means [35, 36]. It adds the concept of fuzzy membership of fuzzy sets, and lower and upper approximations of rough sets into c-means algorithm. While the membership of fuzzy sets enables efficient handling of overlapping partitions, the rough sets deal with uncertainty, vagueness, and incompleteness in class definition. An initialization method is reported based on Pearson's correlation coefficient to select initial cluster centers, along with a parameter optimization technique based on Dunn's cluster validity index.

8.3.1 Rough–Fuzzy C-Means

In rough–fuzzy c-means [35, 36], each cluster β_i is represented by a centroid v_i, a crisp lower approximation $\underline{A}(\beta_i)$, and a fuzzy boundary $B(\beta_i)$. The lower approximation influences the fuzziness of final partition. According to the definitions of lower approximations and boundary of rough sets, if an object $x_j \in \underline{A}(\beta_i)$, then $x_j \notin \underline{A}(\beta_k), \forall k \neq i$, and $x_j \notin B(\beta_i), \forall i$. That is, the object x_j is contained in cluster β_i definitely. Thus, the weights of the objects in lower approximation of a cluster should be independent of other centroids and clusters, and should not be coupled with their similarity with respect to other centroids. Also, the objects in lower approximation of a cluster should have similar influence on the corresponding centroid and cluster. Whereas, if $x_j \in B(\beta_i)$, then the object x_j possibly belongs to β_i and potentially belongs to another cluster. Hence, the objects in boundary regions should have different influence on the centroids and clusters. So, in rough–fuzzy c-means, the membership values of objects in lower approximation are $\mu_{ij} = 1$, while those in boundary region are the same as fuzzy c-means. In other word, the rough–fuzzy

c-means first partitions the data into two classes: lower approximation and boundary. Only the objects in boundary are fuzzified.

The rough–fuzzy c-means algorithm partitions a set of n objects into c clusters by minimizing the following objective function

$$J_{\mathrm{RF}} = \begin{cases} \omega \times \mathscr{A}_1 + (1-\omega) \times \mathscr{B}_1 & \text{if } \underline{A}(\beta_i) \neq \emptyset,\ B(\beta_i) \neq \emptyset \\ \mathscr{A}_1 & \text{if } \underline{A}(\beta_i) \neq \emptyset,\ B(\beta_i) = \emptyset \\ \mathscr{B}_1 & \text{if } \underline{A}(\beta_i) = \emptyset,\ B(\beta_i) \neq \emptyset \end{cases} \tag{8.23}$$

where

$$\mathscr{A}_1 = \sum_{i=1}^{c} \sum_{x_j \in \underline{A}(\beta_i)} ||x_j - v_i||^2; \tag{8.24}$$

and

$$\mathscr{B}_1 = \sum_{i=1}^{c} \sum_{x_j \in B(\beta_i)} (\mu_{ij})^{\acute{m}_1} ||x_j - v_i||^2. \tag{8.25}$$

The parameter ω corresponds to the relative importance of lower and boundary region. Note that μ_{ij} has the same meaning of membership as that in fuzzy c-means. Solving (8.23) with respect to μ_{ij}, we get

$$\mu_{ij} = \left(\sum_{k=1}^{c} \left(\frac{d_{ij}^2}{d_{kj}^2} \right)^{\frac{1}{\acute{m}_1 - 1}} \right)^{-1}; \tag{8.26}$$

where

$$d_{ij}^2 = ||x_j - v_i||^2. \tag{8.27}$$

The new centroid is calculated based on the weighting average of the crisp lower approximation and fuzzy boundary. The centroid calculation for rough–fuzzy c-means is obtained by solving (8.23) with respect to v_i:

$$v_i^{\mathrm{RF}} = \begin{cases} \omega \times \mathscr{C}_1 + (1-\omega) \times \mathscr{D}_1 & \text{if } \underline{A}(\beta_i) \neq \emptyset,\ B(\beta_i) \neq \emptyset \\ \mathscr{C}_1 & \text{if } \underline{A}(\beta_i) \neq \emptyset,\ B(\beta_i) = \emptyset \\ \mathscr{D}_1 & \text{if } \underline{A}(\beta_i) = \emptyset,\ B(\beta_i) \neq \emptyset \end{cases} \tag{8.28}$$

$$\mathscr{C}_1 = \frac{1}{|\underline{A}(\beta_i)|} \sum_{x_j \in \underline{A}(\beta_i)} x_j; \tag{8.29}$$

where $|\underline{A}(\beta_i)|$ represents the cardinality of $\underline{A}(\beta_i)$, and

$$\mathscr{D}_1 = \frac{1}{n_i} \sum_{x_j \in B(\beta_i)} (\mu_{ij})^{\acute{m}_1} x_j; \tag{8.30}$$

where

$$n_i = \sum_{x_j \in B(\beta_i)} (\mu_{ij})^{\acute{m}_1}. \tag{8.31}$$

Thus, the cluster prototypes or centroids depend on the parameter ω and fuzzifier $\acute{m}_1$ rule their relative influence. The correlated influence of both weight parameter and fuzzifier makes it somewhat difficult to determine their optimal values. Since the objects lying in lower approximation definitely belong to a cluster, they are assigned a higher weight ω compared to $(1 - \omega)$ of the objects lying in boundary region. Hence, for rough–fuzzy c-means, the value is given by $0.5 < \omega < 1$.

Approximate optimization of J_{RF} (8.23) by the rough–fuzzy c-means is based on Picard iteration through (8.26) and (8.28). The process starts by randomly choosing c objects as the centroids of the c clusters. The fuzzy memberships of all the objects are calculated using (8.26). Let $u_i = (\mu_{i1}, \ldots, \mu_{ij}, \ldots, \mu_{in})$ be the fuzzy cluster β_i associated with the centroid v_i. After computing μ_{ij} for c clusters and n objects, the values of μ_{ij} for each object x_j are sorted and the difference of two highest memberships of x_j is compared with a threshold value δ. Let μ_{ij} and μ_{kj} be the highest and second highest memberships of x_j. If $(\mu_{ij} - \mu_{kj}) > \delta$, then $x_j \in \underline{A}(\beta_i)$, otherwise $x_j \in B(\beta_i)$ and $x_j \in B(\beta_k)$. After assigning each object in lower approximations or boundary regions of different clusters based on δ, membership value μ_{ij} of the objects are modified. The values of μ_{ij} are set to 1 for the objects in lower approximations, while those in boundary regions are remain unchanged. The new centroids of the clusters are calculated as per (8.28). The main steps of the rough–fuzzy c-means algorithm proceed as follows:

1. Assign initial centroids v_i, $i = 1, 2, \ldots, c$. Choose values for fuzzifier $\acute{m}_1$, and calculate threshold δ. Set iteration counter $t = 1$.
2. Compute μ_{ij} by (8.26) for c clusters and n objects.
3. If μ_{ij} and μ_{kj} be the two highest memberships of x_j and $(\mu_{ij} - \mu_{kj}) \leq \delta$, then $x_j \in B(\beta_i)$ and $x_j \in B(\beta_k)$. Furthermore, x_j is not part of any lower bound.
4. Otherwise, $x_j \in \underline{A}(\beta_i)$.
5. Modify μ_{ij} considering lower and boundary regions for c clusters and n objects.
6. Compute new centroid as per (8.28).
7. Repeat steps two to six, by incrementing t, until no more new assignments can be made.

The performance of rough–fuzzy c-means depends on the value of δ, which determines the class labels of all the objects. In other word, the rough–fuzzy c-means partitions the data set into two classes: lower approximation and boundary, based on the value of δ. The δ represents the size of granules of rough–fuzzy clustering. In practice, the following definition works well:

$$\delta = \frac{1}{n}\sum_{j=1}^{n}(\mu_{ij} - \mu_{kj}) \tag{8.32}$$

where n is the total number of objects, μ_{ij} and μ_{kj} are the highest and second highest memberships of x_j. That is, the value of δ represents the average difference of two highest memberships of all the objects in the data set. A good clustering procedure should make the value of δ as high as possible.

8.3.2 Initialization Method

A limitation of any c-means algorithm is that it can only achieve a local optimum solution that depends on the initial choice of the cluster prototype. Consequently, computing resources may be wasted in that some initial centers get stuck in regions of the input space with a scarcity of data points and may, therefore, never have the chance to move to new locations where they are needed. To overcome this limitation of the c-means algorithm, next the method proposed by Maji and Paul [39] is described to select initial cluster prototype, which is based on a similarity measure using Pearson's correlation coefficient. It enables the algorithm to converge to an optimum or near optimum solutions.

Prior to describe the new method for selecting initial centers, Pearson's correlation coefficient is described next to quantify similarity between two objects. It is the ratio between the covariance of two vectors (x_i, x_j) of expression values of two objects and product of their standard deviations and is given by

$$\boldsymbol{\rho}(x_i, x_j) = \frac{Cov(x_i, x_j)}{\boldsymbol{\sigma}_{x_i}\boldsymbol{\sigma}_{x_j}}; \tag{8.33}$$

that is,

$$\boldsymbol{\rho}(x_i, x_j) = \frac{\sum_{k=1}^{m}(x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)}{\sqrt{\sum_{k=1}^{m}(x_{ik} - \bar{x}_i)^2}\sqrt{\sum_{k=1}^{m}(x_{jk} - \bar{x}_j)^2}}, \tag{8.34}$$

where $\bar{x}_i$ and $\bar{x}_j$ are the means of x_{ik} and x_{jk}, respectively. It considers each gene as a random variable with m observations and measures the similarity between the two genes by calculating the linear relationship between the distributions of the two corresponding random variables.

Based on the concept of Pearson's correlation, next the method is described for selecting initial cluster centers. The main steps of this method proceed as follows:

1. For each object x_i, calculate $\boldsymbol{\rho}(x_j, x_i)$ between itself and the object x_j, $\forall_{j=1}^{n}$.

2. Calculate similarity score between objects x_i and x_j as

$$S(x_j, x_i) = \begin{cases} 1 & \text{if } |\boldsymbol{\rho}(x_j, x_i)| > \lambda \\ 0 & \text{otherwise} \end{cases} \tag{8.35}$$

   where $0.5 \leq \lambda \leq 1$.
3. Calculate total number of similar objects of x_i as

$$N(x_i) = \sum_{j=1}^{n} S(x_j, x_i). \tag{8.36}$$

4. Sort n objects according to their values of $N(x_i)$ such that $N(x_1) > N(x_2) > \ldots > N(x_n)$.
5. If $N(x_i) > N(x_j)$ and $\boldsymbol{\rho}(x_j, x_i) > \lambda$, then x_j cannot be considered as an initial center, resulting in a reduced object set to be considered for initial cluster centers.

Finally, c initial centers are selected from the reduced set as potential initial centers. The main motive of introducing this initialization method lies in identifying different dense regions present in the data set. The identified dense regions ultimately lead to discover natural groups present in data set. The whole approach is, therefore, data dependent.

8.3.3 Identification of Optimum Parameters

The threshold λ in (8.35) plays an important role to generate the initial cluster centers. It controls the degree of similarity among the genes present in microarray data. In effect, it has a direct influence on the performance of the initialization method.

Also, the parameter ω has an influence on the performance of rough–fuzzy c-means algorithm. Since the objects lying in lower approximation definitely belong to a cluster, they are assigned a higher weight ω compared to $(1 - \omega)$ of the objects lying in boundary regions. On the other hand, the performance of the rough–fuzzy c-means significantly reduces when $\omega \simeq 1.00$. In this case, since the clusters cannot see the objects of boundary regions, the mobility of the clusters and the centroids reduces. As a result, some centroids get stuck in local optimum. Hence, to have the clusters and the centroids a greater degree of freedom to move, $0.5 < \omega < 1$.

Let $\mathscr{S} = \{\lambda, \omega\}$ be the set of parameters and $\mathscr{S}^\star = \{\lambda^\star, \omega^\star\}$ is the set of optimal parameters. To find out the optimum set $\mathscr{S}^\star$, containing optimum values of $\lambda^\star$ and $\omega^\star$, the Dunn's cluster validity index [16], reported in Sect. 8.2.2, is used. For each microarray data set, the value of λ is varied from 0.50 to 1.0, while the value of ω is varied from 0.51 to 0.99. The optimum values of $\lambda^\star$ and $\omega^\star$ for each microarray data set is obtained using the following relation:

$$\mathscr{S}^\star = \arg \max_{\mathscr{S}} \{\text{Dunn Index}\}. \tag{8.37}$$

8.4 Experimental Results

In this section, the performance of different algorithms, namely, hard c-means (HCM) [25], fuzzy c-means (FCM) [6], possibilistic c-means (PCM) [30], rough–fuzzy c-means (RFCM) [36, 35], self organizing map (SOM) [53], and CLICK [51] is presented for clustering functionally similar genes from five yeast microarray data sets. The major metrics for evaluating the performance of different algorithms are Silhouette index [48], Davies-Bouldin (DB) index [10], Dunn index [16], and Eisen plot [17]. Also, the biological significance of the gene clusters generated by different methods is analyzed using the gene ontology Term Finder [7, 50]. For each microarray gene expression data set, the number of gene clusters c is decided by using the CLICK [51] algorithm. The input parameters used, which are held constant across all runs, are the values of fuzzifiers $\acute{m}_1 = 2.0$ and $\acute{m}_2 = 2.0$. All the algorithms are implemented in C language and run in LINUX environment having machine configuration Pentium D, 2.66 GHz, 2 MB cache, and 4 GB RAM. The source code of the RFCM algorithm, written in C language, is available at http://www.isical.ac.in/~bibl/results/rfpcm/rfpcm.html.

8.4.1 Gene Expression Data Sets Used

In this chapter, publicly available five yeast microarray gene expression data sets are used to compare the performance of different gene clustering methods. This section gives a brief description of the following five yeast microarray data sets, which are downloaded from Gene Expression Omnibus (http://www.ncbi.nlm.nih.gov/geo/).

8.4.1.1 GDS608

It is a temporal analysis of wild type diploid cells shifted from yeast-form growth in SHAD liquid (plentiful glucose and ammonium) to filamentous-form growth on SLAD agar (low ammonium). The filamentous-form cells were collected hourly for 10 h. The number of genes and time points of this data are 6303 and 10, respectively [47].

8.4.1.2 GDS759

This data set is related to analysis of gene expression in temperature sensitive pre-mRNA splicing factor mutants prp17 null, prp17-1, and prp22-1 at various time points following a shift from the permissive temperature of 23 °C to the restrictive temperature of 37 °C. The number of genes and time points of this data are 6350 and 24, respectively [49].

8.4.1.3 GDS2003

It is the analysis of catabolite-derepressed (galactose) wildtype JM43 and isogenic msn2/4 mutant KKY8 cells shifted to short-term anaerobiosis (2 generations). The Msn2 and 4 are key stress factors. The number of genes and time points are 5617 and 30, respectively [32].

8.4.1.4 GDS2267

It contains the analysis of nutrient-limited continuous-culture cells at twelve 25 min intervals for three cycles. The cells grown under such conditions exhibit robust, periodic cycles in the form of respiratory bursts. The number of genes and time points are 9275 and 36, respectively [55].

8.4.1.5 GDS2712

It represents analysis of *Saccharomyces cerevisiae* BY4743 cells subjected to controlled air-drying and subsequent rehydration (I) for up to 360 min. The data contains 9275 genes and 21 time points [52].

8.4.2 Optimum Values of Different Parameters

Figure 8.1 presents the variation of Dunn index with respect to different values of λ and ω for the RFCM algorithm on GDS608, GDS759, GDS2003, GDS2267, and GDS2715 data sets. From the results reported in Fig. 8.1, it is seen that as the threshold λ increases, the Dunn index value also increases and attains its maximum value at a particular value of $\lambda^{\star}$. After that the Dunn index value decreases with the increase in the value of λ. On the other hand, the Dunn index attains higher values for higher values of ω.

In case of the RFCM algorithm, the optimum values of λ obtained using (8.37) are 1.00, 0.95, 0.85, 0.95, and 1.00 for GDS608, GDS759, GDS2003, GDS2267, and GDS2712 data sets, respectively, while the optimum values of ω obtained using (8.37) are 0.95 for GDS759 and 0.99 for all other data sets. On the other hand, the optimum values of λ for the HCM, obtained using (8.37), are 0.55, 0.90, 0.85, 0.65, and 0.95 for GDS608, GDS759, GDS2003, GDS2267, and GDS2712 data sets, respectively, while the optimum values of λ for the FCM, obtained using (8.37), are 0.95, 0.85, 1.00, 0.95, and 0.50 for GDS608, GDS759, GDS2003, GDS2267, and GDS2712 data sets, respectively. For all data sets, the PCM has failed to produce desired number of gene clusters as it generates coincident clusters even when it has been initialized with the final prototypes of the FCM.

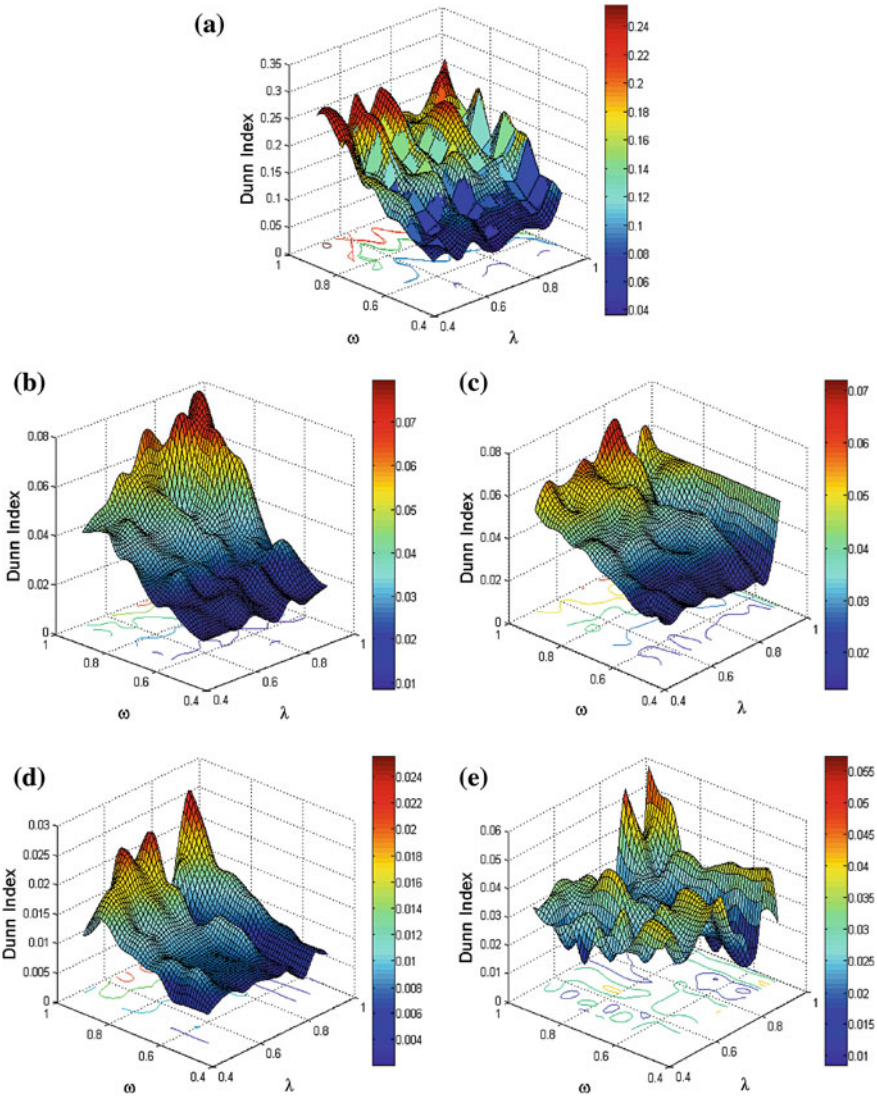

Fig. 8.1 Variation of Dunn index for different values of threshold λ and weight parameter ω. **a** GDS608, **b** GDS759, **c** GDS2003, **d** GDS2267, **e** GDS2712

8.4.3 Importance of Correlation-Based Initialization Method

Table 8.1 provides the comparative results of different c-means algorithms with random initialization of centroids and correlation-based initialization method described in Sect. 8.3.2 for five yeast microarray data sets. The results of each c-means clustering algorithm are reported for their optimal λ and ω values. In most of the cases, the

Table 8.1 Performance analysis of random and correlation-based initialization methods

Data Sets	Initial Centers	Silhouette index			DB index			Dunn index		
		HCM	FCM	RFCM	HCM	FCM	RFCM	HCM	FCM	RFCM
GDS608	Random	0.078	0.005	0.110	1.931	2.082	1.608	0.256	0.000	0.272
	Correlation based	0.080	−0.003	0.101	1.983	2.865	1.850	0.278	0.000	0.251
GDS759	Random	0.082	0.017	0.121	2.392	2.898	1.779	0.035	0.000	0.081
	Correlation based	0.130	−0.004	0.150	2.135	2.615	1.968	0.070	0.000	0.078
GDS2003	Random	0.082	0.014	0.128	2.033	2.975	1.672	0.045	0.000	0.056
	Correlation based	0.073	−0.158	0.102	1.973	2.486	1.957	0.050	0.000	0.069
GDS2267	Random	0.230	0.197	0.233	0.888	2.402	1.006	0.011	0.008	0.016
	Correlation based	0.237	0.194	0.249	0.859	2.591	0.743	0.015	0.008	0.025
GDS2712	Random	0.250	0.208	0.251	0.806	1.760	0.790	0.031	0.012	0.038
	Correlation based	0.251	0.193	0.255	0.795	1.952	0.734	0.032	0.012	0.053

Table 8.2 Performance of different C-means algorithms

Microarray Data sets	Silhouette index			DB index			Dunn index		
	HCM	FCM	RFCM	HCM	FCM	RFCM	HCM	FCM	RFCM
GDS608	0.080	−0.003	0.101	1.983	2.865	1.850	0.278	0.000	0.254
GDS759	0.130	−0.004	0.150	2.135	2.615	1.968	0.070	0.000	0.078
GDS2003	0.073	0.028	0.102	1.973	2.486	1.957	0.050	0.000	0.069
GDS2267	0.237	0.194	0.249	0.859	2.401	0.743	0.015	0.008	0.025
GDS2712	0.251	0.193	0.255	0.795	1.952	0.734	0.032	0.012	0.057

correlation-based initialization method is found to improve the performance in terms of Silhouette index, DB index, and Dunn index for all c-means algorithms. Out of 45 comparisons, the correlation-based initialization method is found to provide significantly better results in 32 cases compared to the random initialization method. From Table 8.1, it is seen that the RFCM algorithm with correlation-based initialization method performs better in eight cases than any other c-means clustering algorithms irrespective of the initialization method.

However, it can also be seen that the HCM algorithm with the correlation-based initialization method outperforms the RFCM algorithm with random initialization method in three cases in terms of Silhouette index and in one case each for DB index and Dunn index, respectively. The better performance of the correlation-based initialization method is achieved due to the fact that it enables the algorithm to converge to an optimum or near optimum solutions.

Table 8.3 Performance of CLICK, SOM, and RFCM

Microarray	Silhouette index			DB Index			Dunn index		
Data sets	CLICK	SOM	RFCM	CLICK	SOM	RFCM	CLICK	SOM	RFCM
GDS608	−0.040	−0.030	0.101	11.520	18.030	1.85	0.060	0.020	0.254
GDS759	−0.080	−0.020	0.15	27.910	19.030	1.968	0.020	0.010	0.078
GDS2003	−0.090	−0.060	0.102	17.610	15.220	1.957	0.050	0.010	0.069
GDS2267	−0.420	0.020	0.249	759.380	5.760	0.743	0.000	0.000	0.025
GDS2712	−0.420	0.070	0.255	293.570	2.070	0.734	0.000	0.000	0.057

8.4.4 Performance Analysis of Different C-Means Algorithms

Table 8.2 presents the performance of different c-means algorithms for their optimum values of λ and ω. For all data sets, the PCM has failed to produce desired number of gene clusters as it generates coincident clusters even when it has been initialized with the final prototypes of the FCM. The results and subsequent discussions are presented with respect to Silhouette index, DB index, and Dunn index. From Table 8.2, it is seen that the RFCM generates better result in most of the cases. However, only for GDS608 data set, the HCM generates better result in terms of Dunn index than that of the FCM and RFCM. All the results establish the fact that the RFCM algorithm is superior to other c-means clustering algorithms. The better performance of the RFCM, in terms of Silhouette, DB, and Dunn indices, is achieved due to the following reasons:

1. the fuzzy membership function of the RFCM algorithm handles efficiently overlapping partitions; and
2. the concept of crisp lower bound and fuzzy boundary of the RFCM algorithm deals with uncertainty, vagueness, and incompleteness in class definition.

8.4.5 Comparative Performance of CLICK, SOM, and RFCM

Table 8.3 presents the comparative performance of CLICK, SOM, and RFCM in terms of Silhouette, DB, and Dunn indices. The RFCM generates better results in all the cases in terms of different cluster validity indices. All the results reported in this table establish the fact that the RFCM algorithm can identify compact groups of coexpressed genes.

8.4.6 Eisen Plots

In Eisen plot [17], the expression value of a gene at a specific time point is represented by coloring the corresponding cell of the data matrix with a color similar to the original color of its spot on the microarray. The shades of red color represent higher

expression level, the shades of green color represent low expression level and the colors toward black represent absence of differential expression values. In the present representation, the genes are ordered before plotting so that the genes that belong to the same cluster are placed one after another. The cluster boundaries are identified by white colored blank rows.

The Eisen plot gives a visual representation of the clustering result. The clustering results produced by the HCM, FCM, and RFCM algorithms on four yeast microarray time-series data sets are visualized by TreeView software, which is available at http://rana.lbl.gov/EisenSoftware and reported in Figs. 8.2 and 8.3. Here in each subfigure, the first image represents the Eisen plots of clusters obtained by the HCM algorithm, while the second and third images are generated by the FCM and RFCM algorithms, respectively. From the Eisen plots presented in Figs. 8.2 and 8.3, it is evident that the expression profiles of the genes in a cluster are similar to each other and they produce similar color pattern, whereas the genes from different clusters differ in color patterns. Also, the results obtained by the RFCM algorithm are more promising than that obtained by both HCM and FCM algorithms.

8.4.7 Biological Significance Analysis

To interpret the biological significance of the generated clusters, the gene ontology (GO) Term Finder is used [7]. It finds the most significantly enriched GO terms associated with the genes belonging to a cluster. The GO project aims to build tree structures and controlled vocabularies, also called ontologies, that describe gene products in terms of their associated biological processes (BPs), molecular functions (MFs), or cellular components (CCs).

The GO term finder determines whether any GO term annotates a specified list of genes at a frequency greater than that would be expected by chance, calculating the associated p-value by using the hypergeometric distribution and the Bonferroni multiple-hypothesis correction [7, 50]:

$$\mathrm{p} = 1 - \sum_{i=0}^{k-1} \frac{\binom{\mathscr{M}}{i}\binom{\mathscr{N}-\mathscr{M}}{n-i}}{\binom{\mathscr{N}}{n}} \tag{8.38}$$

where $\mathscr{N}$ is the total number of genes in the background distribution, $\mathscr{M}$ is the number of genes within that distribution that are annotated, either directly or indirectly, to the node of interest, n is the size of the list of genes of interest and k is the number of genes within that list which are annotated to the node. The closer the p-value is to zero, the more significant the particular GO term associated with the group of genes is, that is, the less likely the observed annotation of the particular GO term to a group of genes occurs by chance.

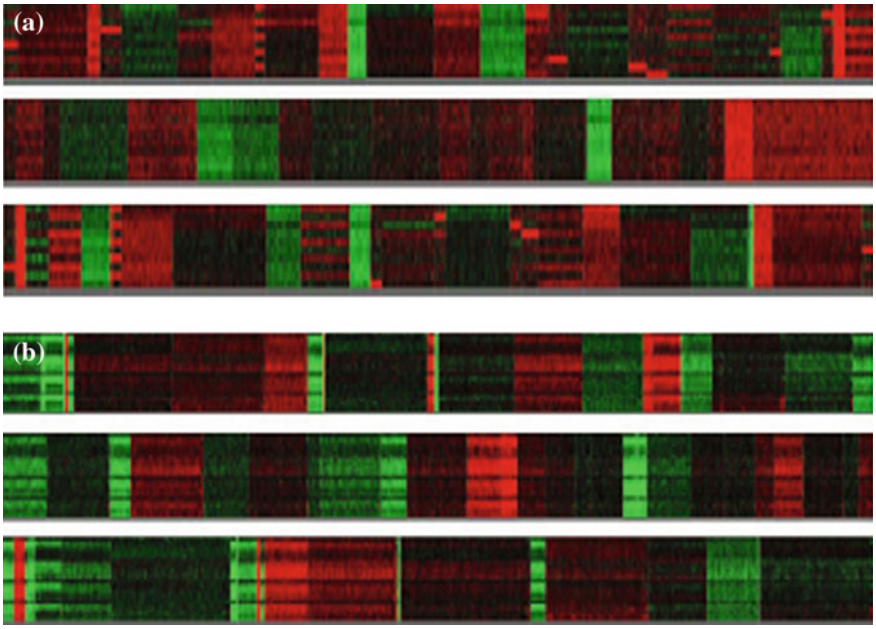

Fig. 8.2 Eisen plots of different clusters for GDS608 and GDS759 (*top* to *bottom*: HCM, FCM and RFCM). **a** GDS608 ($n = 6303$, $m = 10$, $c = 26$), **b** GDS759 ($n = 6350$, $m = 24$, $c = 25$)

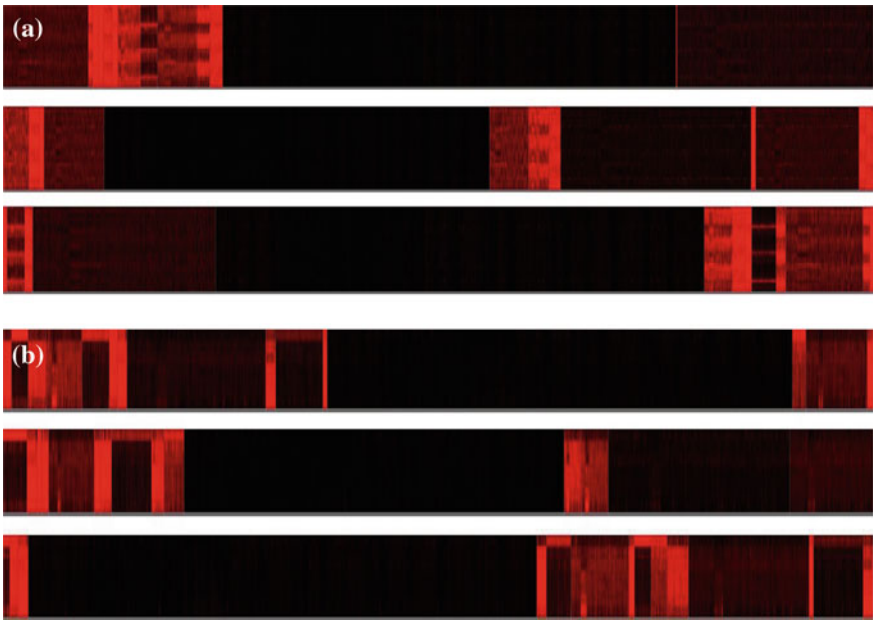

Fig. 8.3 Eisen plots of different clusters for GDS2267 and GDS2712 (*top* to *bottom*: HCM, FCM and RFCM). **a** GDS2267 ($n = 9275$, $m = 36$, $c = 14$), **b** GDS2712 ($n = 9275$, $m = 21$, $c = 15$)

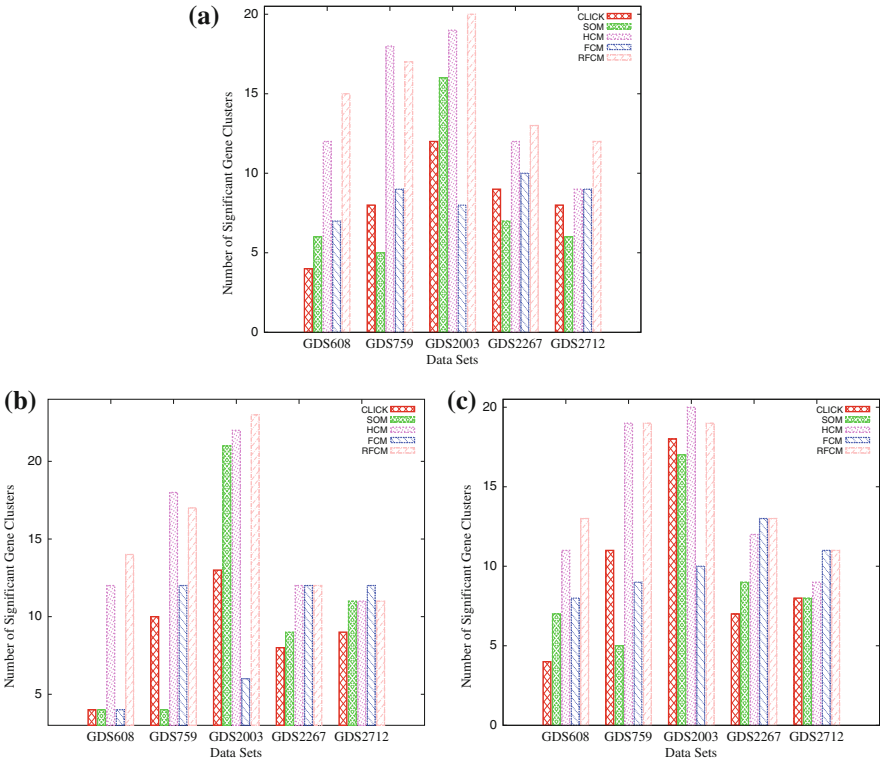

Fig. 8.4 Number of significant gene clusters generated by different algorithms. **a** Molecular function, **b** biological process, **c** cellular component

The GO term finder is used to determine the statistically significant gene clusters produced by the CLICK, SOM, HCM, FCM, and RFCM algorithms for all the GO terms from the BP, MF, and CC ontology. If any cluster of genes generates a p-value smaller than 0.05, then that cluster is considered as a significant cluster. Figure 8.4 presents the number of significant clusters generated by different clustering algorithms for the MF, BP, and CC for all data sets. From Fig. 8.4, it is seen that the RFCM outperforms other clustering algorithms in most of the cases. The RFCM generates more or comparable number of significant gene clusters in most of the cases. However, the HCM generates more number of significant gene clusters in one case each for three ontologies, that is, MF, BP, and CC, respectively. On the other hand, the FCM generates more number of significant gene clusters in one case for BP ontology. All the results reported in Fig. 8.4 establish the fact that rough sets and fuzzy sets-based RFCM algorithm discovers groups of coexpressed genes more efficiently.

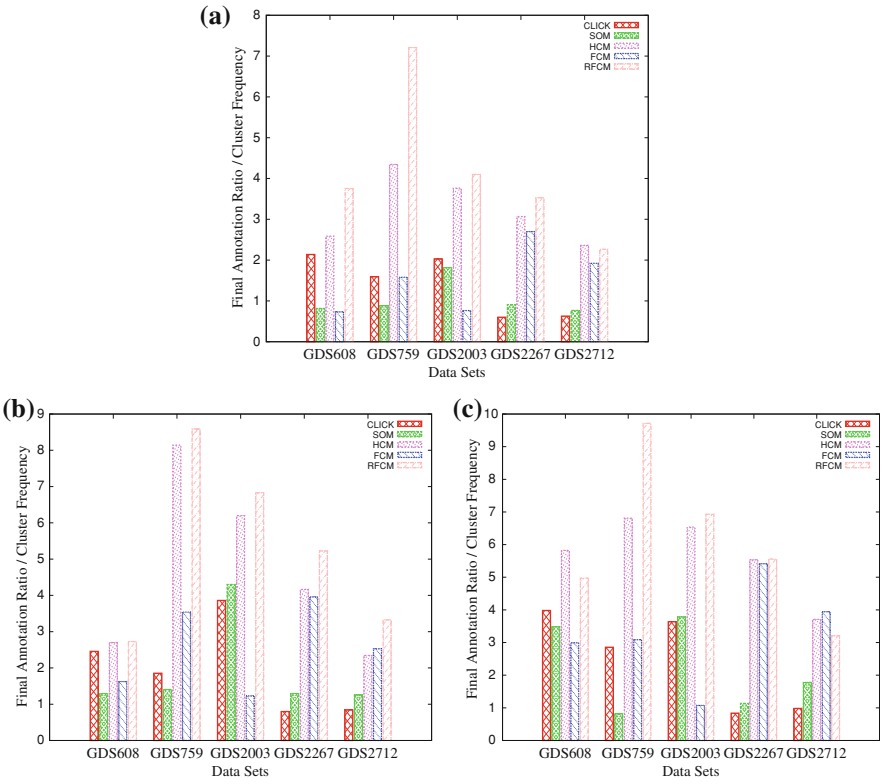

Fig. 8.5 Biological annotation ratios obtained using different algorithms on five gene expression data sets. **a** Molecular function, **b** biological process, **c** cellular component

8.4.8 Functional Consistency of Clustering Result

In order to evaluate the functional consistency of the gene clusters produced by different algorithms, the biological annotations of the gene clusters are considered in terms of the GO. The annotation ratios of each gene cluster in three GO ontologies are calculated using the GO Term Finder [7]. The GO term is searched in which most of the genes of a particular cluster are enriched. The annotation ratio, also termed as cluster frequency, of a gene cluster, is defined as the number of genes in both the assigned GO term and the cluster divided by the number of genes in that cluster. A higher value of annotation ratio indicates that the majority of genes in the cluster are functionally more closer to each other, while a lower value signifies that the cluster contains much more noises or irrelevant genes. After computing the annotation ratios of all gene clusters for a particular ontology, the sum of all annotation ratios is treated as the final annotation ratio. A higher value of final annotation ratio represents that

the corresponding clustering result is better than other, that is, the genes are better clustered by function, indicating a more functionally consistent clustering result [56].

Figure 8.5 presents the comparative results of different gene clustering algorithms, in terms of final annotation ratio or cluster frequency, for the MF, BP, and CC ontologies on five yeast microarray data sets. All the results reported here confirm that the RFCM provides higher or comparable final annotation ratios than that obtained using other clustering algorithms in most of the cases. However, the HCM provides higher final annotation ratios than the RFCM in one and two cases for the MF and CC ontology, respectively. On the other hand, the FCM generates higher final annotation ratio than the RFCM only in one case for the CC ontology.

8.5 Conclusion and Discussion

This chapter addresses the problem of clustering functionally similar genes from microarray time-series gene expression data sets. Different existing gene clustering algorithms have been discussed, along with their merits and demerits. Finally, the application of a new partitive clustering algorithm, termed as rough–fuzzy *c*-means, has been reported to group functionally similar genes. An initialization method is reported, based on Pearson's correlation coefficient, which is found to be successful in effectively circumventing the initialization and local minima problems of iterative refinement clustering algorithms like *c*-means. The effectiveness of rough–fuzzy *c*-means, along with a comparison with existing clustering algorithms, is demonstrated on five yeast microarray data sets. The analysis to identify optimum values of different parameters narrows down the search space and generates better results in terms of different cluster validity indices. The extensive experimental results show that rough–fuzzy *c*-means algorithm produces better clustering results than do the conventional algorithms in terms of Silhouette index, DB index, Dunn index, Eisen plots, number of biologically significant gene clusters, and final annotation ratio.

The algorithms described in this chapter group genes according to similarity measures computed from the gene expressions, without using any information about the response variables. The information of response variables may be incorporated in gene clustering to find groups of coregulated genes with strong association to the response variables. In this regard, next chapter addresses another important task of gene expression data sets, namely, supervised gene clustering, to reveal various groups of coregulated genes with strong association to the response variables.

References

1. Asharaf S, Murty MN (2004) A rough fuzzy approach to web usage categorization. Fuzzy Sets Syst 148:119–129
2. Asharaf S, Shevade SK, Murty MN (2005) Rough support vector clustering. Pattern Recogn 38:1779–1783

3. Au WH, Chan KCC, Wong AKC, Wang Y (2005) Attribute clustering for grouping, selection, and classification of gene expression data. IEEE/ACM Trans Computat Biol Bioinf 2(2):83–101
4. Belacel N, Cuperlovic-Culf M, Laflamme M, Ouellette R (2004) Fuzzy J-means and VNS methods for clustering genes from microarray data. Bioinformatics 20(11):1690–1701
5. Ben-Dor A, Shamir R, Yakhini Z (1999) Clustering gene expression patterns. J Comput Biol 6(3–4):281–297
6. Bezdek JC (1981) Pattern recognition with fuzzy objective function algorithm. Plenum Press, New York
7. Boyle EI, Weng S, Gollub J, Jin H, Botstein D, Cherry JM, Sherlock G (2004) GO:term finder open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. Bioinformatics 20(18):3710–3715
8. Brazma A, Vilo J (2000) Minireview: gene expression data analysis. Fed Eur Biochem Soc Lett 480(1):17–24
9. Causton H, Quackenbush J, Brazma A (2003) Microarray gene expression data analysis: a beginner's guide. Wiley-Blackwell, Oxford
10. Davies DL, Bouldin DW (1979) A cluster separation measure. IEEE Trans Pattern Anal Mach Intell 1:224–227
11. De SK (2004) A rough set theoretic approach to clustering. Fundamenta Informaticae 62(3–4):409–417
12. Dembele D, Kastner P (2003) Fuzzy C-means method for clustering microarray data. Bioinformatics 19(8):973–980
13. D'haeseleer P, Wen X, Fuhrman S, Somogyi R (1998) Mining the gene expression matrix: inferring gene relationships from large scale gene expression data. In: Proceedings of the 2nd international workshop on information processing in cell and tissues, pp 203–212
14. Domany E (2003) Cluster analysis of gene expression data. J Stat Phys 110(3–6):1117–1139
15. Dougherty ER, Barrera J, Brun M, Kim S, Cesar RM, Chen Y, Bittner M, Trent JM (2002) Inference from clustering with application to gene-expression microarrays. J Comput Biol 9:105–126
16. Dunn JC (1974) A fuzzy relative of the ISODATA process and its use in detecting compact, well-separated clusters. J Cybern 3:32–57
17. Eisen MB, Spellman PT, Brown PO, Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. Proc Nat Acad Sci U S A 95(25):14863–14868
18. Fraley C, Raftery AE (1998) How many clusters? which clustering method? answers via model-based cluster analysis. Comput J 41(8):578–588
19. Gasch AP, Eisen MB (2002) Exploring the conditional coregulation of yeast gene expression through fuzzy K-means clustering. Genome Biol 3(11):1–22
20. Ghosh D, Chinnaiyan AM (2002) Mixture modelling of gene expression data from microarray experiments. Bioinformatics 18:275–286
21. Hartuv E, Shamir R (2000) A clustering algorithm based on graph connectivity. Inf Process Lett 76(4–6):175–181
22. Herrero J, Valencia A, Dopazo J (2001) A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics 17:126–136
23. Heyer LJ, Kruglyak S, Yooseph S (1999) Exploring expression data: identification and analysis of coexpressed genes. Genome Res 9(11):1106–1115
24. Hirano S, Tsumoto S (2003) An indiscernibility-based clustering method with iterative refinement of equivalence relations: rough clustering. J Adv Comput Intell Intell Inf 7(2):169–177
25. Jain AK, Dubes RC (1988) Algorithms for clustering data. Prentice Hall, Englewood Cliffs
26. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv 31(3):264–323
27. Jiang, D, Pei J, Zhang A (2003) DHC: a density-based hierarchical clustering method for time-series gene expression data. In: Proceedings of the 3rd IEEE international symposium on bioinformatics and bioengineering, pp 393–400
28. Jiang D, Tang C, Zhang A (2004) Cluster analysis for gene expression data: a survey. IEEE Trans Knowl Data Eng 16(11):1370–1386

29. Klebanov L, Yakovlev A (2007) How high is the level of technical noise in microarray data? Biol Direct 2(9)
30. Krishnapuram R, Keller JM (1993) A possibilistic approach to clustering. IEEE Trans Fuzzy Syst 1(2):98–110
31. Krishnapuram R, Keller JM (1996) The possibilistic C-means algorithm: insights and recommendations. IEEE Trans Fuzzy Syst 4(3):385–393
32. Lai LC, Kosorukoff AL, Burke PV, Kwast KE (2005) Dynamical remodeling of the transcriptome during short-term anaerobiosis in *Saccharomyces cerevisiae*: differential response and role of Msn2 and/or Msn4 and other factors in galactose and glucose media. Mol Cell Biol 25(10):4075–4091
33. Lingras P, West C (2004) Interval set clustering of web users with rough K-means. J Intell Inf Syst 23(1):5–16
34. Maji P (2011) Fuzzy-rough supervised attribute clustering algorithm and classification of microarray data. IEEE Trans Syst Man Cybern Part B Cybern 41(1):222–233
35. Maji P, Pal SK (2007) RFCM: a hybrid clustering algorithm using rough and fuzzy sets. Fundamenta Informaticae 80(4):475–496
36. Maji P, Pal SK (2007) Rough set based generalized fuzzy C-means algorithm and quantitative indices. IEEE Trans Syst Man Cybern Part B Cybern 37(6):1529–1540
37. Maji P, Pal SK (2012) Rough-fuzzy pattern recognition: applications in bioinformatics and medical imaging. Wiley-IEEE Computer Society Press, New Jersey
38. Maji P, Paul S (2011) Microarray time-series data clustering using rough-fuzzy C-means algorithm. In: Proceedings of the 5th IEEE international conference on bioinformatics and biomedicine, Atlanta, pp 269–272
39. Maji P, Paul S (2013) Robust rough-fuzzy C-means algorithm: design and applications in coding and non-coding RNA expression data clustering. Fundamenta Informaticae 124:153–174
40. Maji P, Paul S (2013) Rough-fuzzy clustering for grouping functionally similar genes from microarray data. IEEE/ACM Trans Comput Biol Bioinf 10(2):286–299
41. McLachlan GJ, Bean RW, Peel D (2002) A mixture model-based approach to the clustering of microarray expression data. Bioinformatics 18:413–422
42. McLachlan GJ, Do KA, Ambroise C (2004) Analyzing microarray gene expression data. John Wiley and Sons, Hoboken
43. McQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley symposium on mathematics, statistics and probability, pp 281–297
44. Pal SK, Gupta BD, Mitra P (2004) Rough self organizing map. Appl Intell 21(3):289–299
45. Pal SK, Mitra P (2002) Multispectral image segmentation using the rough set-initialized-EM algorithm. IEEE Trans Geosci Remote Sens 40(11):2495–2501
46. Pawlak Z (1991) Rough sets: theoretical aspects of resoning about data. Kluwer, Dordrecht
47. Prinz S, Avila-Campillo I, Aldridge C, Srinivasan A, Dimitrov K, Siegel AF, Galitski T (2004) Control of yeast filamentous-form growth by modules in an integrated molecular network. Genome Res 14(3):380–390
48. Rousseeuw JP (1987) Silhouettes: a graphical aid to the interpration and validation of cluster analysis. J Comput Appl Math 20:53–65
49. Sapra AK, Arava Y, Khandelia P, Vijayraghavan U (2004) Genome-wide analysis of pre-mRNA splicing: intron features govern the requirement for the second-step factor, Prp17 in *Saccharomyces cerevisiae* and *Schizosaccharomyces pombe*. J Biol Chem 279(50):52437–52446
50. Sevilla JL, Segura V, Podhorski A, Guruceaga E, Mato JM, Martinez-Cruz LA, Corrales FJ, Rubio A (2005) Correlation between gene expression and GO semantic similarity. IEEE/ACM Trans Comput Biol Bioinf 2(4):330–338
51. Shamir R, Sharan R (2000) CLICK: a clustering algorithm for gene expression analysis. In: Proceedings of the 8th international conference on intelligent systems for, molecular biology, pp 307–31

52. Singh J, Kumar D, Ramakrishnan N, Singhal V, Jervis J, Garst JF, Slaughter SM, DeSantis AM, Potts M, Helm RF (2005) Transcriptional response of *Saccharomyces cerevisiae* to desiccation and rehydration. Appl Environ Microbiol 71(12):8752–8763
53. Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander ES, Golub TR (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. Proc Nat Acad Sci U S A 96(6):2907–2912
54. Tavazoie S, Hughes D, Campbell MJ, Cho RJ, Church GM (1999) Systematic determination of genetic network architecture. Nat Genet 22(3):281–285
55. Tu BP, Kudlicki A, Rowicka M, McKnight SL (2005) Logic of the yeast metabolic cycle: temporal compartmentalization of cellular processes. Science 310(5751):1152–1158
56. Wang H, Wang Z, Li X, Gong B, Feng L, Zhou Y (2011) A robust approach based on weibull distribution for clustering gene expression data. Algorithms Mol Biol 6(1):14
57. Woolf PJ, Wang Y (2000) A fuzzy logic approach to analyzing gene expression data. Physiol Genomics 3:9–15
58. Xing EP, Karp RM (2001) CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. Bioinformatics 17(1):306–315
59. Yeung KY, Fraley C, Murua A, Raftery AE, Ruzz WL (2001) Model-based clustering and data transformations for gene expression data. Bioinformatics 17(10):977–987
60. Zadeh LA (1965) Fuzzy sets. Inf Control 8:338–353

Chapter 9
Mutual Information Based Supervised Attribute Clustering for Microarray Sample Classification

9.1 Introduction

Recent advancement and wide use of high-throughput technology are producing an explosion in using gene expression phenotype for identification and classification in a variety of diagnostic areas. An important application of gene expression data in functional genomics is to classify samples according to their gene expression profiles [9, 13]. For most gene expression data, the number of training samples is generally very small compared to the large number of genes involved in the experiments. When the number of genes is significantly greater than the number of samples, it is possible to find biologically relevant correlations of gene behavior with the sample categories or response variables [27].

However, among the large amount of genes, only a small fraction is effective for performing a certain task. Also, a small subset of genes are desirable in developing gene expression-based diagnostic tools for delivering precise, reliable, and interpretable results [43]. With the gene selection results, the cost of biological experiment and decision can be greatly reduced by analyzing only the marker genes. Hence, identifying a reduced set of most relevant genes is the goal of gene selection . The small number of training samples and a large number of genes make gene selection a more relevant and challenging problem in gene expression-based classification. As this is a feature selection problem [6, 24, 25], the clustering method can be used, which partitions the given gene set into subgroups, each of which should be as homogeneous as possible [2, 10, 20, 21, 41].

When applied to gene expression data analysis, the conventional clustering methods such as Bayesian clustering [22, 34], hierarchical clustering [14, 17], k-means algorithm [18], self-organizing map [14, 40], and principal component analysis [33, 45] group a subset of genes that are interdependent or correlated with each other. In other words, genes or attributes in a cluster are more correlated with each other, whereas genes in different clusters are less correlated [2, 21, 41]. The attribute clustering is able to reduce the search dimension of a classification algorithm and constructs the model using a tightly correlated subset of genes rather than using the

P. Maji and S. Paul, *Scalable Pattern Recognition Algorithms*,
DOI: 10.1007/978-3-319-05630-2_9,

entire gene space. After clustering genes, a reduced set of genes can be selected for further analysis [2, 21, 41]. A brief survey on various gene clustering algorithms is reported in Chap. 8.

However, all these algorithm group genes according to unsupervised similarity measures computed from the gene expressions, without using any information about the sample categories or response variables. The information of response variables should be incorporated in attribute clustering to find groups of co-regulated genes with strong association to the sample categories [5]. In this background, some supervised attribute clustering algorithms such as supervised gene clustering [5], gene shaving [16], tree harvesting [15], and partial least square procedure [35] have been reported to reveal groups of co-regulated genes with strong association to the sample categories. The supervised attribute clustering is defined as the grouping of genes or attributes, controlled by the information of sample categories or response variables.

In general, the quality of generated clusters is always relative to a certain criterion. Different criteria may lead to different clustering results. However, every criterion tries to measure the similarity among the subset of genes present in a cluster. While tree harvesting [15] uses an unsupervised similarity measure to group a set of co-regulated genes, other supervised algorithms such as supervised gene clustering [5], gene shaving [16], and partial least square procedure [35] do not use any similarity measure to cluster genes; rather use different predictive scores such as Wilcoxon test [5] and Cox model score test [16] to measure gene-class relevance. Moreover, all these measures depend on the actual values of the training data. Hence, they may be sensitive to noise or outlier of the data set [8, 18, 21, 36]. On the other hand, as mutual information [3, 8, 19, 36] depends only on the probability distribution of a random variable, it has been widely used for computing both gene-class relevance and gene-gene redundancy or similarity [2, 3, 7, 8, 19, 28, 36].

This chapter presents a mutual information-based supervised attribute clustering (MISAC) algorithm [31] to find co-regulated clusters of genes whose collective expression is strongly associated with the sample categories or class labels. A new quantitative measure, based on mutual information, is introduced to compute the similarity between attributes. The new measure incorporates the information of sample categories while measuring the similarity between attributes. In effect, it helps to identify functional groups of genes that are of special interest in sample classification. The new supervised attribute clustering method uses this measure to reduce the redundancy among genes. It involves partitioning of the original gene set into some distinct subsets or clusters so that the genes within a cluster are highly co-regulated with strong association to the sample categories while those in different clusters are as dissimilar as possible. A single gene from each cluster having the highest gene-class relevance value is first selected as the initial representative of that cluster. The representative of each cluster is then modified by averaging the initial representative with other genes of that cluster whose collective expression is strongly associated with the sample categories. Finally, the modified representative of each cluster is selected to constitute the resulting reduced feature set. In effect, the MISAC algorithm yields biologically significant gene clusters, whose coherent average expression levels allow perfect discrimination of sample categories.

Also, the MISAC algorithm avoids the noise sensitivity problem of existing supervised gene clustering algorithms. The performance of the MISAC algorithm, along with a comparison with existing algorithms, is studied both qualitatively and quantitatively on three cancer and two arthritis data sets using the class separability index and the predictive accuracy of naive Bayes classifier, K-nearest neighbor rule, and support vector machine.

The structure of the rest of this chapter is as follows: Sect. 9.2 briefly introduces existing supervised and unsupervised gene clustering algorithms, along with different existing criteria used for computing the relevance and redundancy. The new supervised attribute clustering algorithm is presented in Sect. 9.3. A few case studies and a comparison with existing algorithms are presented in Sect. 9.4. Concluding remarks are given in Sect. 9.5.

9.2 Clustering Genes for Sample Classification

In this section, some existing supervised and unsupervised gene clustering algorithms are reported, along with different widely used criteria for computing gene-class relevance and gene-gene redundancy.

9.2.1 Gene Clustering: Supervised Versus Unsupervised

Clustering is one of the major tasks in gene expression data analysis. To find groups of co-regulated genes from microarray data, different unsupervised clustering techniques such as hierarchical clustering [14, 17], k-means algorithm [18], self organizing map [14, 40], and principal component analysis [33, 45] have been widely used. The hierarchical clustering identifies sets of correlated genes with similar behavior across the samples, but yields thousands of clusters in a tree-like structure, which makes the identification of functional groups very difficult [14, 17]. In contrast, self organizing map [14, 40] and k-means algorithm [18] require a prespecified number and an initial spatial structure of clusters, but this may be hard to come up with in real problems. However, these algorithms usually fail to reveal functional groups of genes that are of special interest in sample classification as the genes are clustered by similarity only, without using any information about the sample categories or class labels [5].

To reveal groups of co-regulated genes with strong association to the sample categories, different supervised attribute clustering algorithms have been proposed recently [5, 15, 16, 35]. One notable work in this field encompasses tree harvesting [15], a two step method which consists first of generating numerous candidate groups by unsupervised hierarchical clustering. Then, the average expression profile of each cluster is considered as a potential input variable for a response model and the few gene groups that contain the most useful information for tissue discrimination are

identified. Only this second step makes the clustering supervised, as the selection process relies on external information about the tissue types. Another supervised clustering method, called gene shaving, identifies subsets of genes with coherent expression patterns and large variation across the conditions [16]. The technique can be unsupervised, where the genes and samples are treated as unlabeled, or partially or fully supervised by using known properties of the genes or samples to assist in finding meaningful groupings.

An interesting supervised clustering approach that directly incorporates the response variables in the grouping process is the partial least squares procedure [35], which in a supervised manner constructs weighted linear combinations of genes that have maximal covariance with the outcome. However, it has the drawback that the fitted components involve all (usually thousands of) genes, which makes them very difficult to interpret. Moreover, partial least squares for every component yields a linear combination of gene expressions which completely lacks the biological interpretation of having a cluster of genes acting similarly in the same pathway.

A direct approach to combine gene selection, clustering, and supervision in one single step is reported in [5]. A similar single step approach is also pursued by Jornsten and Yu [23]. The supervised attribute clustering algorithm proposed in [5] is a combination of gene selection for cluster membership and formation of a new predictor by possible sign flipping and averaging the gene expressions within a cluster. The cluster membership is determined with a forward and backward searching technique that optimizes the Wilcoxon test-based predictive score and margin criterion defined in [5], which both involve the supervised response variables from the data. However, as both predictive score and margin criterion depend on the actual gene expression values, they are very much sensitive to noise or outlier of the data set.

9.2.2 Criteria for Gene Selection and Clustering

As reported in Chap. 5, the t-test, F-test [8, 26], information gain, mutual information [8, 36], normalized mutual information [28], and f-information [29] are typically used to measure the relevance of a gene with respect to the class labels or sample categories and the same or a different metric such as mutual information, the L_1 distance, Euclidean distance, and Pearson's correlation coefficient [8, 21, 36] is employed to calculate the similarity or redundancy between genes.

To measure the relevance of a gene, the t-test is widely used, assuming that there are two classes of samples in a gene expression data set. When there are multiple classes of samples, the t-test is typically computed for one class versus all the other classes. For multiple classes of samples, an F-test between a gene and the class label can be used to calculate the relevance score of that gene. The F-test reduces to the t-test for two class problem with the relation $F = t^2$. In [5], the Wilcoxon's test statistic is used to compute the relevance of a gene assuming two classes of samples in microarray data set.

On the other hand, the Euclidean distance measures the difference in the individual magnitudes of each gene. However, the genes regarded as similar by the Euclidean distance may be very dissimilar in terms of their shapes. Similarly, the Euclidean distance between two genes having an identical shape may be large if they differ from each other by a large scaling factor. But, the overall shapes of genes are of the primary interest for gene expression data [21]. Hence, the Euclidean distance may not be able to yield a good proximity measurement of genes [21]. The Pearson's correlation coefficient considers each gene as a random variable and measures the similarity between two genes by calculating the linear relationship between distributions of two corresponding random variables. An empirical study has shown that Pearson's correlation coefficient is not robust to outliers and it may assign high similarity score to a pair of dissimilar genes [18].

However, as the t-test, F-test, Wilcoxon's test, Euclidean distance, and Pearson's correlation depend on the actual gene expression values of microarray data, they are very much sensitive to noise or outlier of the data set. On the other hand, as the information theoretic measure such as entropy, mutual information, and f-information depends only on the probability distribution of a random variable rather than on its actual values, it is more effective to evaluate the gene-class relevance as well as gene-gene redundancy [8, 29, 36].

In principle, the mutual information is used to quantify the information shared by two objects. If two independent objects do not share much information, the mutual information value between them is small. While two highly correlated objects will demonstrate a high mutual information value [39]. The objects can be the class label and the genes. The necessity for a gene to be an independent and informative can, therefore, be determined by the shared information between the gene and the rest as well as the shared information between the gene and class label [8, 36]. If a gene has expression values randomly or uniformly distributed in different classes, its mutual information with these classes is zero. If a gene is strongly differentially expressed for different classes, it should have large mutual information. Thus, the mutual information can be used as a measure of relevance of genes. Similarly, the mutual information may be used to measure the level of similarity or redundancy between two genes.

9.3 Supervised Gene Clustering Algorithm

In this section, a mutual information-based supervised attribute clustering (MISAC) algorithm [31] is presented for grouping co-regulated genes with strong association to the class labels. It is based on a supervised similarity measure that follows next.

9.3.1 Supervised Similarity Measure

In real-data analysis, one of the important issues is computing both relevance and redundancy of attributes by discovering dependencies among them. Intuitively, a set

of attributes $\mathbb{Q}$ depends totally on a set of attributes $\mathbb{P}$, if all attribute values from $\mathbb{Q}$ are uniquely determined by values of attributes from $\mathbb{P}$. If there exists a functional dependency between values of $\mathbb{Q}$ and $\mathbb{P}$, then $\mathbb{Q}$ depends totally on $\mathbb{P}$.

Let $\mathbb{U} = \{x_1, \ldots, x_i, \ldots, x_n\}$ be the set of n samples and $\mathbb{A} = \{\mathscr{A}_1, \ldots, \mathscr{A}_i, \ldots, \mathscr{A}_m\}$ denotes the set of m attributes of a given data set $\mathscr{T} = \{w_{ij} | i = 1, \ldots, m, j = 1, \ldots, n\}$, where $w_{ij} \in \Re$ is the measured value of the attribute $\mathscr{A}_i$ in the sample x_j. Let $\mathbb{D} = \{D_1, \ldots, D_i, \ldots, D_n\}$ be the set of class labels or sample categories of n samples. Define $\mathrm{R}_{\mathscr{A}_i}(\mathbb{D})$ as the relevance of the attribute $\mathscr{A}_i$ with respect to the class label $\mathbb{D}$ while $\mathrm{S}(\mathscr{A}_i, \mathscr{A}_j)$ as the redundancy or similarity between two attributes $\mathscr{A}_i$ and $\mathscr{A}_j$. The mutual information can be used to calculate both relevance and redundancy among attributes.

The relevance $\mathrm{R}_{\mathscr{A}_i}(\mathbb{D})$ of the attribute $\mathscr{A}_i$ with respect to the class label $\mathbb{D}$ using mutual information can be calculated as follows:

$$\mathrm{R}_{\mathscr{A}_i}(\mathbb{D}) = I(\mathscr{A}_i, \mathbb{D}) \tag{9.1}$$

where $I(\mathscr{A}_i, \mathbb{D})$ represents the mutual information between attribute $\mathscr{A}_i$ and class label $\mathbb{D}$ that is given by

$$I(\mathscr{A}_i, \mathbb{D}) = H(\mathscr{A}_i) - H(\mathscr{A}_i|\mathbb{D}). \tag{9.2}$$

Here, $H(\mathscr{A}_i)$ and $H(\mathscr{A}_i|\mathbb{D})$ represent the entropy of attribute $\mathscr{A}_i$ and the conditional entropy of $\mathscr{A}_i$ given class label $\mathbb{D}$, respectively. The entropy $H(\mathscr{A}_i)$ is known to be a measure of the amount of uncertainty about the attribute $\mathscr{A}_i$, while $H(\mathscr{A}_i|\mathbb{D})$ is the amount of uncertainty left in $\mathscr{A}_i$ when knowing $\mathbb{D}$. Hence, the quantity $I(\mathscr{A}_i, \mathbb{D})$ is the reduction in the uncertainty of the attribute $\mathscr{A}_i$ by the knowledge of class label $\mathbb{D}$. In other words, it represents the amount of information that the class label $\mathbb{D}$ contains about the attribute $\mathscr{A}_i$.

Definition 9.1 For continuous random variables such as gene expression values, the entropy, conditional entropy, and mutual information can be defined as follows:

$$H(\mathscr{Y}) = -\int p(y) \log p(y) \mathrm{d}y; \tag{9.3}$$

$$H(\mathscr{Y}|\mathscr{Z}) = -\int p(y, z) \log p(y|z) \mathrm{d}y \mathrm{d}z; \tag{9.4}$$

$$I(\mathscr{Y}, \mathscr{Z}) = \int\int p(y, z) \log \frac{p(y, z)}{p(y)p(z)} \mathrm{d}y \mathrm{d}z \tag{9.5}$$

where $p(y)$ is the true probability density function of the attribute or variable $\mathscr{Y}$, while $p(y|z)$ and $p(y, z)$ represent the conditional probability density function of $\mathscr{Y}$ given the variable $\mathscr{Z}$ and the joint probability density function of $\mathscr{Y}$ and $\mathscr{Z}$,

respectively. Usually, the Gaussian function is used to approximate the true density function [12].

The redundancy or similarity between two attributes $\mathscr{A}_i$ and $\mathscr{A}_j$, in terms of mutual information, can also be calculated as follows:

$$\mathrm{S}(\mathscr{A}_i, \mathscr{A}_j) = I(\mathscr{A}_i, \mathscr{A}_j). \tag{9.6}$$

However, the term $\mathrm{S}(\mathscr{A}_i, \mathscr{A}_j)$ does not incorporate the information of sample categories or class labels $\mathbb{D}$ while measuring the similarity and it is considered as unsupervised similarity measure. Hence, a new quantitative measure, called supervised similarity measure, is reported here based on mutual information for measuring the similarity between two random variables. It incorporates the information of sample categories or class labels while measuring the similarity between attributes.

Definition 9.2 The significance of an attribute $\mathscr{A}_j$ with respect to another attribute $\mathscr{A}_i$ can be defined as follows:

$$\sigma_{\{\mathscr{A}_i, \mathscr{A}_j\}}(\mathbb{D}, \mathscr{A}_j) = \mathrm{R}_{\{\mathscr{A}_i, \mathscr{A}_j\}}(\mathbb{D}) - \mathrm{R}_{\mathscr{A}_i}(\mathbb{D}). \tag{9.7}$$

That is, the significance of an attribute $\mathscr{A}_j$ is the change in dependency when the attribute $\mathscr{A}_j$ is removed from the set $\{\mathscr{A}_i, \mathscr{A}_j\}$. The higher the change in dependency, the more significant the attribute $\mathscr{A}_j$ is. If the significance is 0, then the attribute $\mathscr{A}_j$ is dispensable.

Based on the concept of significance of an attribute, the supervised similarity measure between two attributes is defined next.

Definition 9.3 The supervised similarity between two attributes $\mathscr{A}_i$ and $\mathscr{A}_j$ is defined as follows [31]:

$$\Psi(\mathscr{A}_i, \mathscr{A}_j) = \frac{1}{1+\kappa^2}, \tag{9.8}$$

$$\text{where } \kappa = \left\{ \frac{\sigma_{\{\mathscr{A}_i, \mathscr{A}_j\}}(\mathbb{D}, \mathscr{A}_j) + \sigma_{\{\mathscr{A}_i, \mathscr{A}_j\}}(\mathbb{D}, \mathscr{A}_i)}{2} \right\} \tag{9.9}$$

$$\text{that is, } \kappa = \mathrm{R}_{\{\mathscr{A}_i, \mathscr{A}_j\}}(\mathbb{D}) - \left\{ \frac{\mathrm{R}_{\mathscr{A}_i}(\mathbb{D}) + \mathrm{R}_{\mathscr{A}_j}(\mathbb{D})}{2} \right\}. \tag{9.10}$$

Hence, the supervised similarity measure $\Psi(\mathscr{A}_i, \mathscr{A}_j)$ directly takes into account the information of sample categories or class labels $\mathbb{D}$ while computing the similarity between two attributes $\mathscr{A}_i$ and $\mathscr{A}_j$. If attributes $\mathscr{A}_i$ and $\mathscr{A}_j$ are completely correlated with respect to class labels $\mathbb{D}$, then $\kappa = 0$ and so $\Psi(\mathscr{A}_i, \mathscr{A}_j)$ is 1. If $\mathscr{A}_i$ and $\mathscr{A}_j$ are totally uncorrelated, $\Psi(\mathscr{A}_i, \mathscr{A}_j) \to 0$. Hence, $\Psi(\mathscr{A}_i, \mathscr{A}_j)$ can be used as a measure of supervised similarity between two attributes $\mathscr{A}_i$ and $\mathscr{A}_j$. The following properties can be stated about the measure:

1. $0 < \Psi(\mathscr{A}_i, \mathscr{A}_j) \leq 1$.
2. $\Psi(\mathscr{A}_i, \mathscr{A}_j) = 1$ if and only if $\mathscr{A}_i$ and $\mathscr{A}_j$ are completely correlated.
3. $\Psi(\mathscr{A}_i, \mathscr{A}_j) \to 0$ if and only if $\mathscr{A}_i$ and $\mathscr{A}_j$ are totally uncorrelated.
4. $\Psi(\mathscr{A}_i, \mathscr{A}_j) = \Psi(\mathscr{A}_j, \mathscr{A}_i)$ (symmetric).

The supervised similarity between two attributes $\mathscr{A}_i$ and $\mathscr{A}_j$, in terms of entropy, is given by

$$\Psi(\mathscr{A}_i, \mathscr{A}_j) = \Big[1 + \Big[H(\mathscr{A}_i\mathscr{A}_j|\mathbb{D}) - \frac{1}{2}\Big\{H(\mathscr{A}_i|\mathscr{A}_j) + H(\mathscr{A}_j|\mathscr{A}_i) + H(\mathscr{A}_i|\mathbb{D}) + H(\mathscr{A}_j|\mathbb{D})\Big\}\Big]^2\Big]^{-1}. \tag{9.11}$$

Combining (9.6) and (9.11), the term $\Psi(\mathscr{A}_i, \mathscr{A}_j)$ can be expressed as follows:

$$\Psi(\mathscr{A}_i, \mathscr{A}_j) = \Big[1 + \Big[\mathrm{S}(\mathscr{A}_i, \mathscr{A}_j) + H(\mathscr{A}_i\mathscr{A}_j|\mathbb{D}) - \frac{1}{2}\Big\{H(\mathscr{A}_i) + H(\mathscr{A}_j) + H(\mathscr{A}_i|\mathbb{D}) + H(\mathscr{A}_j|\mathbb{D})\Big\}\Big]^2\Big]^{-1}. \tag{9.12}$$

Hence, the supervised similarity measure $\Psi(\mathscr{A}_i, \mathscr{A}_j)$ not only considers the information of sample categories or class labels $\mathbb{D}$, it also takes into account the unsupervised similarity between two attributes $\mathrm{S}(\mathscr{A}_i, \mathscr{A}_j)$.

9.3.2 Gene Clustering Algorithm

The mutual information-based supervised attribute clustering algorithm [31], termed as MISAC, relies on mainly two factors, namely, determining the relevance of each attribute and growing the cluster around each relevant attribute incrementally by adding one attribute after the other. One of the important property of this clustering approach is that the cluster is augmented by the attributes those satisfy following two conditions:

1. suit best into the current cluster in terms of a supervised similarity measure defined above; and
2. improve the differential expression of the current cluster most, according to the relevance of the cluster representative or prototype.

The growth of a cluster is repeated until the cluster stabilizes, and then the MISAC algorithm starts to generate a new cluster.

Let $\mathrm{R}_{\mathscr{A}_i}(\mathbb{D})$ be the relevance of attribute $\mathscr{A}_i \in \mathbb{A}$ with respect to class label $\mathbb{D}$. The relevance uses information about the class labels and is thus a criterion for supervised clustering. The MISAC algorithm starts with a single attribute $\mathscr{A}_i$ that has the highest relevance value with respect to class labels. An initial cluster $\mathbb{V}_i$ is formed by selecting the set of attributes $\{\mathscr{A}_j\}$ from the whole set $\mathbb{A}$ considering the attribute $\mathscr{A}_i$ as the representative of cluster $\mathbb{V}_i$, where

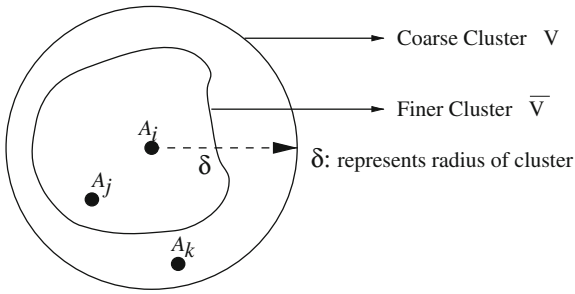

Fig. 9.1 Representation of a supervised attribute cluster

$$\mathbb{V}_i = \{\mathscr{A}_j | \Psi(\mathscr{A}_i, \mathscr{A}_j) \geq \delta; \mathscr{A}_j \neq \mathscr{A}_i \in \mathbb{A}\}. \tag{9.13}$$

Hence, the cluster $\mathbb{V}_i$ represents the set of attributes of $\mathbb{A}$ those have the supervised similarity values with the attribute $\mathscr{A}_i$ greater than a predefined threshold value δ. The cluster $\mathbb{V}_i$ is the coarse cluster corresponding to the attribute $\mathscr{A}_i$, while the threshold δ is termed as the radius of cluster $\mathbb{V}_i$ (Fig. 9.1).

After forming the initial coarse cluster $\mathbb{V}_i$, the cluster representative is refined incrementally. By searching among the attributes of cluster $\mathbb{V}_i$, the current cluster representative is merged and averaged with one single attribute such that the augmented cluster representative $\bar{\mathscr{A}}_i$ increases the relevance value. The merging process is repeated until the relevance value can no longer be improved. Instead of averaging all attributes of $\mathbb{V}_i$, the augmented attribute $\bar{\mathscr{A}}_i$ is computed by considering a subset of attributes $\bar{\mathbb{V}}_i \subset \mathbb{V}_i$ those increase the relevance value of cluster representative $\bar{\mathscr{A}}_i$. The set of attributes $\bar{\mathbb{V}}_i$ represents the finer cluster of the attribute $\mathscr{A}_i$ (Fig. 9.1). While the generation of coarse cluster reduces the redundancy among attributes of the set $\mathbb{A}$, that of finer cluster increases the relevance with respect to class labels. After generating the augmented cluster representative $\bar{\mathscr{A}}_i$ from the finer cluster $\bar{\mathbb{V}}_i$, the process is repeated to find more clusters and augmented cluster representatives by discarding the set of attributes $\mathbb{V}_i$ from the whole set $\mathbb{A}$.

To compute the set $\mathbb{V}_i$ corresponding to the attribute $\mathscr{A}_i$, one may consider the conventional unsupervised similarity measure $\mathrm{S}(\mathscr{A}_i, \mathscr{A}_j)$ as defined in (9.6). However, as it does not take into account the information of sample categories or class labels, the attributes are clustered by similarity only, without using any information about the sample categories. In effect, it fails to reveal functional groups of attributes that are of special interest in classification. On the other hand, as the supervised similarity measure $\Psi(\mathscr{A}_i, \mathscr{A}_j)$ defined in (9.8) incorporates the class information directly while computing the similarity between two attributes $\mathscr{A}_i$ and $\mathscr{A}_j$, it can identify functional groups present in the attribute set.

The main steps of the mutual information-based supervised attribute clustering (MISAC) algorithm are reported next.

- Let $\mathbb{C}$ be the set of attributes of the original data set, while $\mathbb{S}$ and $\bar{\mathbb{S}}$ are the set of actual and augmented attributes, respectively, selected by the MISAC algorithm.

- Let $\mathbb{V}_i$ be the coarse cluster associated with the attribute $\mathscr{A}_i$ and $\bar{\mathbb{V}}_i$, the finer cluster of $\mathscr{A}_i$ (Fig. 9.1), represents the set of attributes of $\mathbb{V}_i$ those are merged and averaged with the attribute $\mathscr{A}_i$ to generate the augmented cluster representative $\bar{\mathscr{A}_i}$.

1. Initialize $\mathbb{C} \leftarrow \mathbb{A} = \{\mathscr{A}_1, \ldots, \mathscr{A}_i, \ldots, \mathscr{A}_j, \ldots, \mathscr{A}_m\}$, $\mathbb{S} \leftarrow \emptyset$, and $\bar{\mathbb{S}} \leftarrow \emptyset$.
2. Calculate the relevance value $\mathrm{R}_{\mathscr{A}_i}(\mathbb{D})$ of each attribute $\mathscr{A}_i \in \mathbb{C}$.
3. Repeat the following nine steps (steps 4–12) until $\mathbb{C} = \emptyset$ or desired number of attributes is selected.
4. Select attribute $\mathscr{A}_i$ from $\mathbb{C}$ as the representative of cluster $\mathbb{V}_i$ that has highest relevance value. In effect, $\mathscr{A}_i \in \mathbb{S}$, $\mathscr{A}_i \in \mathbb{V}_i$, $\mathscr{A}_i \in \bar{\mathbb{V}}_i$, and $\mathbb{C} = \mathbb{C} \setminus \mathscr{A}_i$.
5. Generate coarse cluster $\mathbb{V}_i$ from the set of existing attributes of $\mathbb{C}$ satisfying the following condition:

$$\mathbb{V}_i = \{\mathscr{A}_j | \Psi(\mathscr{A}_i, \mathscr{A}_j) \geq \delta; \mathscr{A}_j \neq \mathscr{A}_i \in \mathbb{C}\}.$$

6. Initialize $\bar{\mathscr{A}_i} \leftarrow \mathscr{A}_i$.
7. Repeat following four steps (steps 8–11) for each attribute $\mathscr{A}_j \in \mathbb{V}_i$.
8. Compute two augmented cluster representatives by averaging $\mathscr{A}_j$ and its complement with the attributes of $\bar{\mathbb{V}}_i$ as follows:

$$\bar{\mathscr{A}}^{+}_{i+j} = \frac{1}{|\bar{\mathbb{V}}_i| + 1} \left\{ \sum_{\mathscr{A}_k \in \bar{\mathbb{V}}_i} \mathscr{A}_k + \mathscr{A}_j \right\}; \tag{9.14}$$

$$\bar{\mathscr{A}}^{-}_{i+j} = \frac{1}{|\bar{\mathbb{V}}_i| + 1} \left\{ \sum_{\mathscr{A}_k \in \bar{\mathbb{V}}_i} \mathscr{A}_k - \mathscr{A}_j \right\}. \tag{9.15}$$

9. The augmented cluster representative $\bar{\mathscr{A}_{i+j}}$ after averaging $\mathscr{A}_j$ or its complement with $\bar{\mathbb{V}}_i$ is as follows:

$$\bar{\mathscr{A}_{i+j}} = \begin{cases} \bar{\mathscr{A}}^{+}_{i+j} & \text{if } \mathrm{R}_{\bar{\mathscr{A}}^{+}_{i+j}}(\mathbb{D}) \geq \mathrm{R}_{\bar{\mathscr{A}}^{-}_{i+j}}(\mathbb{D}) \\ \bar{\mathscr{A}}^{-}_{i+j} & \text{otherwise.} \end{cases} \tag{9.16}$$

10. The augmented cluster representative $\bar{\mathscr{A}_i}$ of cluster $\mathbb{V}_i$ is $\bar{\mathscr{A}_{i+j}}$ if $\mathrm{R}_{\bar{\mathscr{A}_{i+j}}}(\mathbb{D}) \geq \mathrm{R}_{\bar{\mathscr{A}_i}}(\mathbb{D})$, otherwise $\bar{\mathscr{A}_i}$ remains unchanged.
11. Select attribute $\mathscr{A}_j$ or its complement as a member of the finer cluster $\bar{\mathbb{V}}_i$ of attribute $\mathscr{A}_i$ if $\mathrm{R}_{\bar{\mathscr{A}_{i+j}}}(\mathbb{D}) \geq \mathrm{R}_{\bar{\mathscr{A}_i}}(\mathbb{D})$.
12. In effect, $\bar{\mathscr{A}_i} \in \bar{\mathbb{S}}$ and $\mathbb{C} = \mathbb{C} \setminus \mathbb{V}_i$.
13. Sort the set of augmented cluster representatives $\bar{\mathbb{S}} = \{\bar{\mathscr{A}_i}\}$ according to their relevance value $\mathrm{R}_{\bar{\mathscr{A}_i}}(\mathbb{D})$ with respect to the class labels $\mathbb{D}$.
14. Stop.

9.3.3 Fundamental Property

From the above discussions, the following properties corresponding to each cluster $\mathbb{V}_i$ can be derived:

1. $\Psi(\mathscr{A}_i, \mathscr{A}_j) \geq \delta; \forall \mathscr{A}_j \in \mathbb{V}_i$.
2. $\mathrm{R}_{\mathscr{A}_i}(\mathbb{D}) \geq \mathrm{R}_{\mathscr{A}_j}(\mathbb{D}); \forall \mathscr{A}_j \in \mathbb{V}_i$.
3. $\mathrm{R}_{\bar{\mathscr{A}}_{i+j}}(\mathbb{D}) \geq \mathrm{R}_{\bar{\mathscr{A}}_i}(\mathbb{D}); \forall \mathscr{A}_j \in \bar{\mathbb{V}}_i$.
4. $\mathrm{R}_{\bar{\mathscr{A}}_{i+j}}(\mathbb{D}) < \mathrm{R}_{\bar{\mathscr{A}}_i}(\mathbb{D}); \forall \mathscr{A}_j \in \mathbb{V}_i \setminus \bar{\mathbb{V}}_i$.
5. $\mathbb{V}_i \cap \mathbb{V}_k = \emptyset, \forall i \neq k$.

The property 1 says that if an attribute $\mathscr{A}_j \in \mathbb{V}_i \Rightarrow \Psi(\mathscr{A}_i, \mathscr{A}_j) \geq \delta$. That is, the supervised similarity between the attribute $\mathscr{A}_j$ of coarse cluster $\mathbb{V}_i$ and the initial cluster representative $\mathscr{A}_i$ is greater than a predefined threshold value δ. The property 2 establishes the fact that if $\mathscr{A}_j \in \mathbb{V}_i \Rightarrow \mathrm{R}_{\mathscr{A}_i}(\mathbb{D}) \geq \mathrm{R}_{\mathscr{A}_j}(\mathbb{D})$, that is, the relevance of the cluster representative $\mathscr{A}_i$ is the maximum among that of all attributes of the cluster $\mathbb{V}_i$. The properties 3 and 4 are of great importance in increasing the relevance of augmented cluster representative with respect to the class labels and reducing the redundancy among the attribute set. The property 3 says that if $\mathscr{A}_j \in \bar{\mathbb{V}}_i \Rightarrow \mathrm{R}_{\bar{\mathscr{A}}_{i+j}}(\mathbb{D}) \geq \mathrm{R}_{\bar{\mathscr{A}}_i}(\mathbb{D})$. It means an attribute $\mathscr{A}_j$ belongs to the finer cluster $\bar{\mathbb{V}}_i$ if and only if it increases the relevance value of the augmented cluster representative $\bar{\mathscr{A}}_i$. On the other hand, property 4 says that the attributes those belong to only coarse cluster $\mathbb{V}_i$, not to finer cluster $\bar{\mathbb{V}}_i$, are not responsible to increase the relevance of augmented cluster representative. Hence, the set of attributes $\bar{\mathbb{V}}_i$ increases the relevance value of the attribute $\mathscr{A}_i$ as well as reduces the redundancy of the whole set, while the set of attributes $\mathbb{V}_i \setminus \bar{\mathbb{V}}_i$ is only responsible for reducing the redundancy. Finally, property 5 says that if an attribute $\mathscr{A}_i \in \mathbb{V}_i \Rightarrow \mathscr{A}_i \notin \mathbb{V}_k, \forall k \neq i$, that is, the attribute $\mathscr{A}_i$ is contained in $\mathbb{V}_i$ only. Hence, the MISAC algorithm generates nonoverlapping attribute clusters.

9.3.4 Computational Complexity

The computation of the relevance of m attributes is carried out in step 2 of the MISAC algorithm, which has $\mathscr{O}(m)$ time complexity. The cluster generation steps, that is steps 4–12, are executed c times to generate c clusters and corresponding augmented cluster representatives. There are three loops in the cluster generation steps, which are executed m, m, and m_i times, respectively, where $m_i < m$ represents the cardinality of the cluster $\mathbb{V}_i$. Each iteration of the loops takes only a constant amount of time. Hence, the complexity to generate c clusters using steps 4–12 is $\mathscr{O}(c(m+m_i))$. The computing time of $\mathscr{O}(c(m+m_i))$ becomes $\mathscr{O}(cm)$ for any value of m_i. Finally, step 13 performs the sorting of c augmented cluster representatives according to their relevance values, which has a computational complexity of $\mathscr{O}(c^2)$.

Hence, the overall time complexity of the MISAC algorithm is $\mathcal{O}(m + cm + c^2)$, that is, $\mathcal{O}(cm + c^2)$. However, as the number of desired clusters c is constant and sufficiently small compared to the total number of attributes m, the MISAC algorithm has an overall $\mathcal{O}(m)$ time complexity.

9.4 Experimental Results

The performance of the mutual information-based supervised attribute clustering (MISAC) algorithm [31] is extensively compared with that of some existing supervised and unsupervised gene clustering and gene selection algorithms, namely, ACA (attribute clustering algorithm) [2], MBBC (model-based Bayesian clustering) [22], SGCA (supervised gene clustering algorithm) [5], GS (gene shaving) [16], mRMR (minimum redundancy-maximum relevance framework) [8], and the method proposed by Golub et al. [13]. To analyze the performance of different algorithms, the experimentation is done on five microarray gene expression data sets. The major metrics for evaluating the performance of different algorithms are the class separability index [6] and classification accuracy of naive Bayes (NB) classifier [10], K-nearest neighbor (K-NN) rule [10], and support vector machine (SVM) [42], which are briefly described in Chap. 5. To compute the classification accuracy, the leave-one-out cross-validation is performed on each gene expression data set.

The MISAC algorithm is implemented in C language and run in LINUX environment having machine configuration Pentium IV, 3.2 GHz, 1 MB cache, and 1 GB RAM. The kernel-based method is used to approximate probability density functions by combining basis functions [12]. It consists in superposing a Gaussian function to each point of the feature. The final probability density function approximation is obtained by taking the envelope of all the basis functions superposed at each point. The gnu scientific library is used to implement the kernel-based approach.

9.4.1 Gene Expression Data Sets Used

In this chapter, publicly available three cancer and two arthritis data sets are used. Since binary classification is a typical and fundamental issue in diagnostic and prognostic prediction of cancer and arthritis, different methods are compared using five binary-class data sets, namely, breast cancer [44], leukemia [13], colon cancer [1], RAOA [37], and RAHC [38]. Details of breast cancer, leukemia, and colon cancer data sets are reported in Chap. 5. Descriptions of remaining two data sets are presented next.

9.4.1.1 Rheumatoid Arthritis Versus Osteoarthritis (RAOA)

The RAOA data set consists of gene expression profiles of thirty patients: 21 with RA and 9 with OA [37]. The Cy5-labeled experimental cDNA and the Cy3 labeled

common reference sample were pooled and hybridized to the lymphochips containing ∼18,000 cDNA spots representing genes of relevance in immunology [37].

9.4.1.2 Rheumatoid Arthritis Versus Healthy Controls (RAHC)

The RAHC data set consists of gene expression profiling of peripheral blood cells from 32 patients with RA, 3 patients with probable RA, and 15 age and sex-matched healthy controls performed on microarrays with a complexity of ∼26 K unique genes (43 K elements) [38].

9.4.2 Optimum Value of Threshold

The threshold δ in (9.13) plays an important role to form the initial coarse cluster. It controls the degree of similarity among the attributes of a cluster. In effect, it has a direct influence on the performance of the MISAC algorithm. If δ increases, the number of attributes in a cluster decreases, but the similarity among them with respect to sample categories increases. On the other hand, the similarity among the attributes of a cluster decreases with the decrease in the value of δ.

To find out the optimum value of δ, the class separability index $\mathscr{S}$ [6] is used, which is reported in Chap. 5. For five microarray data sets, the value of δ is varied from 0.80 to 1.0 and the class separability index is computed only for best cluster ($c = 1$). Figure 9.2 represents the variation of class separability index with respect to different values of threshold δ on colon cancer, RAHC, and RAOA data sets. From the results reported in Fig. 9.2, it is seen that as the threshold δ increases, the class separability index decreases and attains its minimum value at a particular value of δ. After that the class separability index increases with the increase in the value of δ. Hence, the optimum value of δ for each microarray data set is obtained using the following relation:

$$\delta_{\text{optimum}} = \arg\min_{\delta}\{\mathscr{S}\}. \tag{9.17}$$

The optimum values of δ obtained using (9.17) are 0.97, 0.96, 0.93, 0.98, and 0.96 for breast, leukemia, colon, RAHC, and RAOA data sets, respectively. Finally, Tables 9.1 and 9.2 present the performance of the MISAC algorithm for different values of δ. The results and subsequent discussions are presented with respect to the classification accuracy of the SVM, K-NN rule, and NB classifier. The results are reported for three best clusters ($c = 3$) obtained using the MISAC algorithm. From the results reported in Tables 9.1 and 9.2, it is also seen that the MISAC algorithm achieves its best performance at $\delta = \delta_{\text{optimum}}$, irrespective of the classifiers used. However, the performance of the MISAC algorithm at $\delta = 0.98$ is same as that at δ_{optimum} for RAOA data set with respect to the classification accuracy of three classifiers.

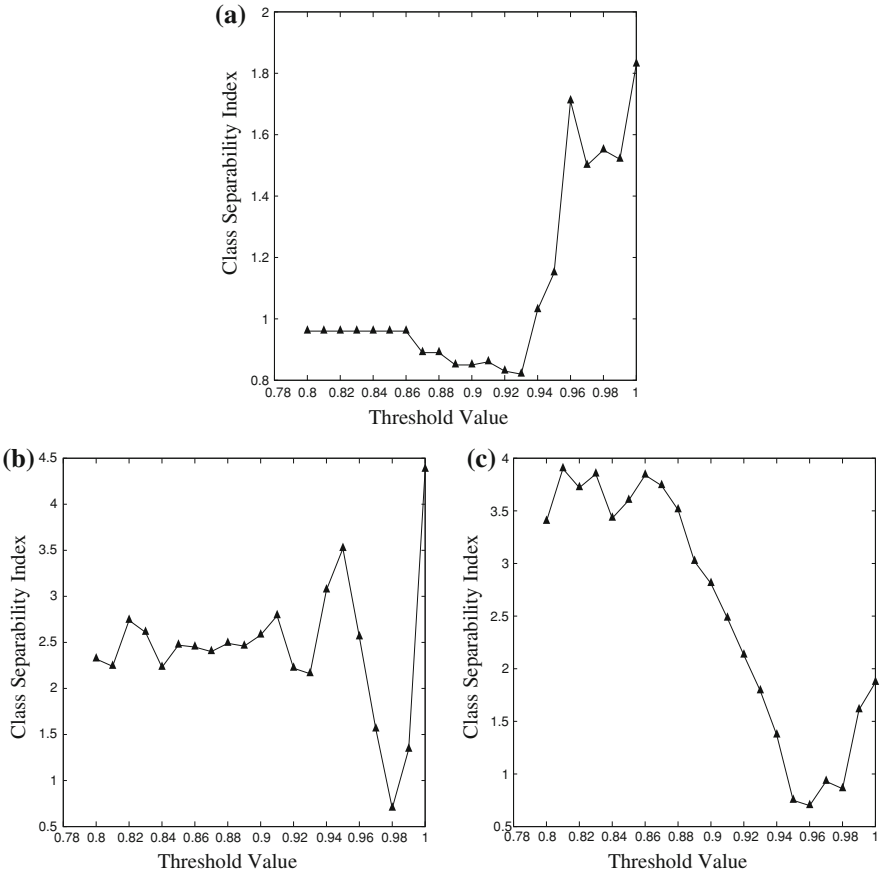

Fig. 9.2 Variation of class separability index for different values of threshold δ **a** Colon **b** RAHC **c** RAOA

9.4.3 Qualitative Analysis of Supervised Clusters

For three cancer and two arthritis data sets, the best clusters generated by the MISAC algorithm are analyzed using the Eisen plot [11]. In Figs. 9.3 and 9.4, the results of best cluster obtained using the MISAC algorithm are reported for colon cancer and RAHC data sets considering the values of δ as 0.93 and 0.98, respectively. Figures 9.3 and 9.4a show the expression values of the actual genes or attributes of the best cluster over the samples for two data sets. Figures 9.3 and 9.4b represent the Eisen plot of corresponding finer cluster with actual gene expression values, while Figs. 9.3 and 9.4c show the expression values of the augmented cluster representatives of the best cluster for two data sets. In Figs. 9.5, 9.6, and 9.7, the expression values of the actual and augmented cluster representatives of the best cluster are presented for breast, leukemia, and RAOA data sets considering δ as 0.97, 0.96, and 0.96, respectively.

Table 9.1 Performance of the MISAC algorithm on three cancer microarray data sets for different values of threshold δ

Data sets	Value of c	Measure	Different values of threshold δ														
			0.85	0.86	0.87	0.88	0.89	0.90	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99
Breast	1	SVM	91.8	89.8	91.8	87.8	91.8	83.7	91.8	93.9	95.9	85.7	91.8	95.9	100	95.9	87.8
		K-NN	98.0	95.9	100	95.9	98.0	95.9	98.0	93.9	98.0	98.0	98.0	95.9	100	95.9	91.8
		NB	100	100	100	100	100	100	100	100	100	100	100	100	100	98.0	91.8
	2	SVM	91.8	91.8	93.9	89.8	91.8	91.8	93.9	100	98.0	98.0	98.0	98.0	100	95.9	87.8
		K-NN	95.9	95.9	93.9	95.9	100	95.9	95.9	100	98.0	95.9	98.0	100	100	95.9	89.8
		NB	100	100	100	100	100	100	100	100	100	100	100	100	100	98.0	89.8
	3	SVM	100	91.8	91.8	91.8	91.8	91.8	93.9	100	98.0	100	100	100	100	95.9	93.9
		K-NN	100	93.9	93.9	95.9	93.9	93.9	95.9	100	100	100	100	100	100	95.9	93.9
		NB	100	100	100	100	100	100	100	100	100	100	100	100	100	98.0	91.8
Leukemia	1	SVM	93.1	91.7	94.4	97.2	97.2	95.8	94.4	95.8	97.2	95.8	97.2	98.6	97.2	98.6	90.3
		K-NN	98.6	98.6	98.6	97.2	98.6	95.8	95.8	98.6	100	100	97.2	100	98.6	98.6	93.1
		NB	100	100	100	100	100	100	100	100	100	100	100	100	100	100	94.4
	2	SVM	91.7	97.2	94.4	98.6	98.6	97.2	95.8	100	98.6	98.6	98.6	100	100	100	94.4
		K-NN	97.2	97.2	98.6	98.6	98.6	97.2	95.8	100	100	98.6	98.6	100	100	100	94.4
		NB	100	100	100	100	100	100	100	100	100	100	100	100	100	100	98.6
	3	SVM	93.1	97.2	93.1	98.6	98.6	100	100	100	100	98.6	100	100	100	100	93.1
		K-NN	97.2	97.2	100	98.6	98.6	98.6	100	100	98.6	98.6	100	100	100	100	94.4
		NB	100	100	100	100	98.6	100	100	100	100	100	100	100	100	100	98.6
Colon	1	SVM	96.8	96.8	100	100	100	96.8	98.4	96.8	100	98.4	88.7	98.4	100	98.4	90.3
		K-NN	98.4	98.4	100	100	96.8	96.8	98.4	98.4	100	98.4	95.2	100	96.8	96.8	90.3
		NB	100	100	100	100	100	100	100	100	100	100	100	100	100	98.4	91.9
	2	SVM	98.4	98.4	100	100	98.4	95.2	98.4	100	100	100	98.4	100	98.4	98.4	90.3
		K-NN	98.4	98.4	98.4	98.4	100	95.2	98.4	100	100	100	100	100	96.8	98.4	90.3
		NB	100	100	100	100	100	100	100	100	100	100	100	100	98.4	98.4	93.6
	3	SVM	100	100	100	100	100	100	98.4	100	100	100	100	100	96.8	96.8	88.7
		K-NN	100	100	100	100	100	100	98.4	100	100	100	100	100	96.8	98.4	91.9
		NB	100	100	100	100	100	100	100	100	100	100	100	100	96.8	96.8	91.9

All the results reported in Figs. 9.3, 9.4, 9.5, 9.6, and 9.7 establish the fact that the MISAC algorithm can identify groups of co-regulated genes with strong association to the sample categories or class labels.

9.4.4 Importance of Supervised Similarity Measure

The supervised similarity measure based on mutual information, defined in (9.8), takes into account the information of sample categories or class labels while computing the similarity between two genes. It also incorporates the unsupervised similarity measure among genes. On the other hand, mutual information-based conventional similarity measure of (9.6) does not consider the class labels or sample categories.

Table 9.2 Performance of the MISAC algorithm on two arthritis microarray data sets for different values of threshold δ

Data sets	Value of c	Measure	Different values of threshold δ														
			0.85	0.86	0.87	0.88	0.89	0.90	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99
RAHC	1	SVM	82.0	78.0	82.0	80.0	78.0	76.0	80.0	82.0	90.0	82.0	70.0	84.0	86.0	100	94.0
		K-NN	96.0	94.0	92.0	98.0	94.0	92.0	96.0	96.0	94.0	92.0	90.0	98.0	98.0	100	94.0
		NB	100	100	100	100	100	100	100	100	100	100	100	100	100	100	94.0
	2	SVM	78.0	98.0	78.0	78.0	78.0	98.0	78.0	98.0	100	88.0	84.0	90.0	100	100	98.0
		K-NN	96.0	98.0	94.0	94.0	94.0	98.0	92.0	98.0	100	94.0	92.0	98.0	100	100	98.0
		NB	100	100	100	100	100	100	100	100	100	100	100	100	100	100	96.0
	3	SVM	98.0	98.0	98.0	98.0	100	98.0	98.0	98.0	100	100	86.0	100	100	100	98.0
		K-NN	96.0	98.0	98.0	96.0	96.0	96.0	100	98.0	100	100	92.0	100	100	100	100
		NB	100	100	100	100	100	100	100	100	100	100	100	100	100	100	96.0
RAOA	1	SVM	73.3	73.3	73.3	76.7	80.0	80.0	76.7	80.0	76.7	76.7	96.7	100	70.0	100	93.3
		K-NN	86.7	96.7	100	93.3	93.3	100	96.7	90.0	96.7	100	96.7	100	83.3	100	96.7
		NB	100	100	100	100	100	100	100	100	100	100	100	100	100	100	96.7
	2	SVM	73.3	76.7	73.3	76.7	80.0	80.0	76.7	80.0	80.0	80.0	96.7	100	96.7	100	93.3
		K-NN	96.7	96.7	93.3	93.3	93.3	90.0	100	86.7	90.0	100	100	100	100	100	93.3
		NB	100	100	100	100	100	100	100	100	100	100	100	100	100	100	96.7
	3	SVM	70.0	70.0	66.7	73.3	76.7	76.7	53.3	83.3	80.0	73.3	96.7	100	96.7	100	96.7
		K-NN	96.7	96.7	93.3	90.0	93.3	90.0	90.0	90.0	100	93.3	100	100	100	100	93.3
		NB	100	100	100	100	100	100	100	100	100	100	100	100	100	100	90.0

In order to establish the importance of supervised similarity measure over existing conventional unsupervised similarity measure, the extensive experimentation is carried out on three cancer and two arthritis data sets. Finally, the best results obtained using unsupervised similarity measure are compared with that of new supervised measure in Table 9.3 with respect to the class separability (CS) index and classification accuracy of the SVM, K-NN rule, and NB classifier. From all the results reported in Table 9.3, it is seen that the performance of the new supervised similarity measure is better compared to that of the unsupervised measure for all microarray data sets. That is, the new supervised similarity measure can identify functional groups of genes present in the microarray, while the unsupervised similarity fails to reveal that. However, the unsupervised similarity measure performs better than supervised similarity with respect to the class separability index for $c = 2$ in case of leukemia and RAHC data sets, and for $c = 3$ in case of leukemia, RAHC, and RAOA data sets.

9.4.5 Importance of Augmented Genes

Each coarse cluster represents the set of genes or attributes those have the supervised similarity values with the initial cluster representative greater than a predefined threshold value δ. In fact, the relevance of the initial cluster representative is greater

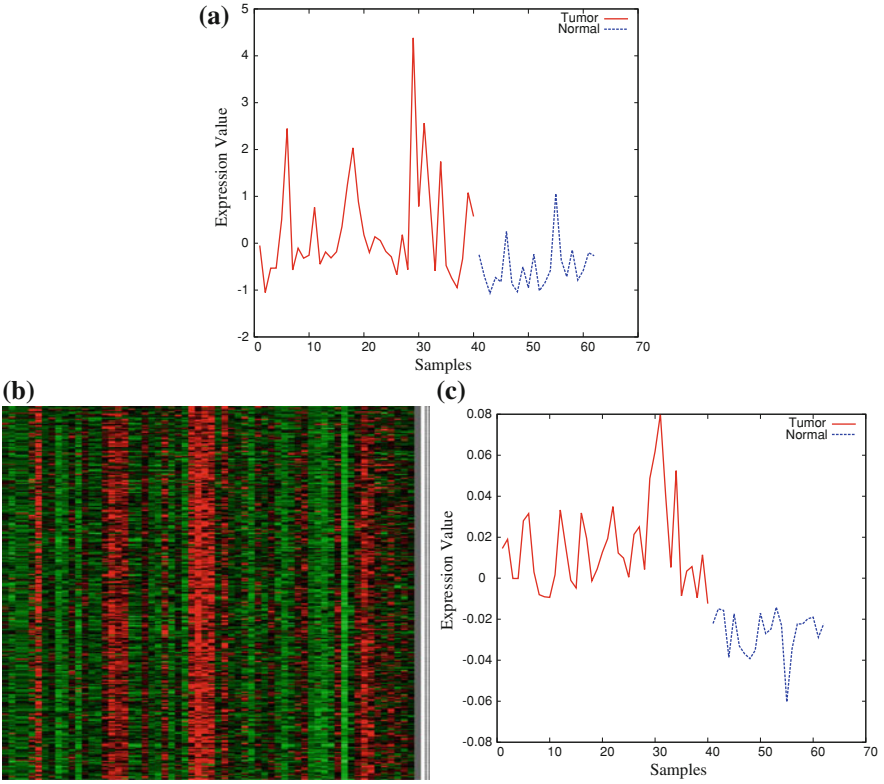

Fig. 9.3 Results obtained using the MISAC algorithm on colon cancer data set for $\delta = 0.93$ **a** Initial expression value **b** Eisen plot **c** Augmented expression value

than that of other genes of that cluster. After forming the initial coarse cluster, the cluster representative is refined incrementally in the MISAC algorithm. By searching among the genes of coarse cluster, the current cluster representative is merged and averaged with one single gene such that the augmented cluster representative increases the relevance value. The merging process is repeated until the relevance value can no longer be improved.

In order to establish the importance of augmented cluster representative over initial cluster representative, that is, actual gene, extensive experiments are carried out on five microarray data sets. Table 9.4 reports the comparative performance of actual and augmented genes of different clusters. Results are reported for $c = 3$ considering supervised similarity measure. The performance of actual and augmented genes is compared with respect to the class separability index and classification accuracy of the SVM, K-NN rule, and NB classifier. All the results reported in Table 9.4 establish the fact that the MISAC algorithm performs significantly better in case of augmented gene than the actual gene. Only in case of leukemia data for $c = 2$ and 3, and RAOA

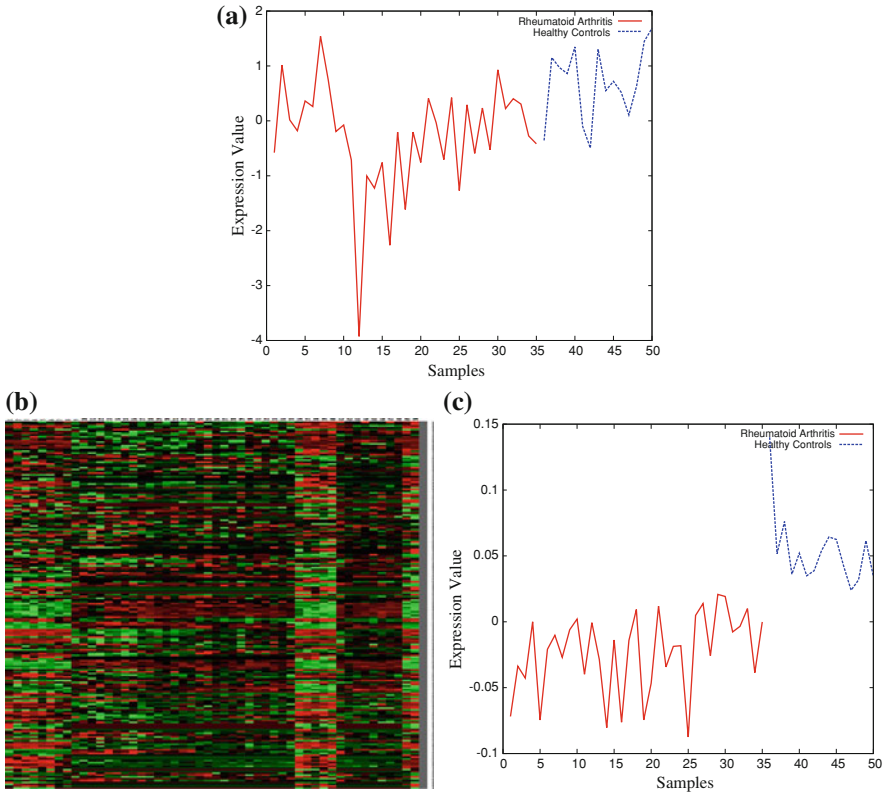

Fig. 9.4 Results obtained using the MISAC algorithm on RAHC data set for $\delta = 0.98$ **a** Initial expression value **b** Eisen plot **c** Augmented expression value

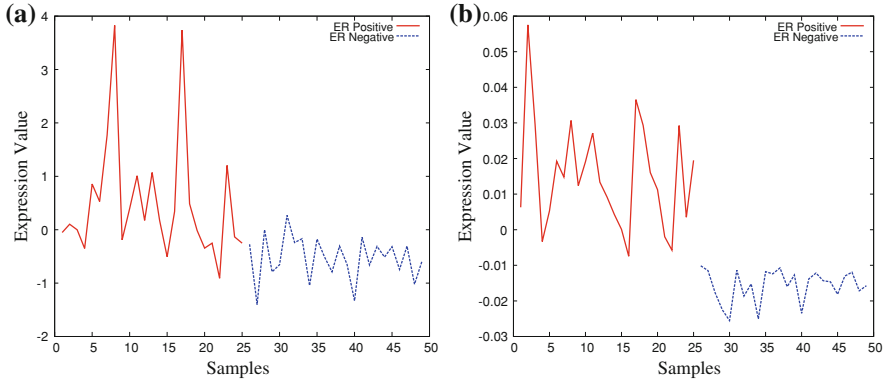

Fig. 9.5 Results obtained for breast cancer data considering $\delta = 0.97$ **a** Initial representative **b** Augmented representative

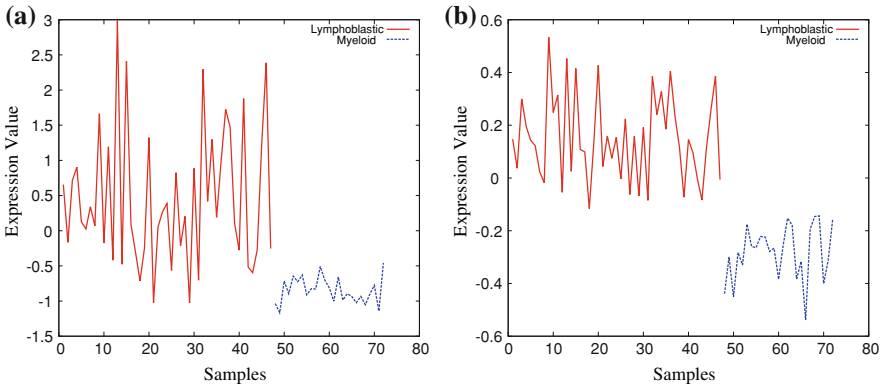

Fig. 9.6 Results obtained for leukemia data considering $\delta = 0.96$ **a** Initial representative **b** Augmented representative

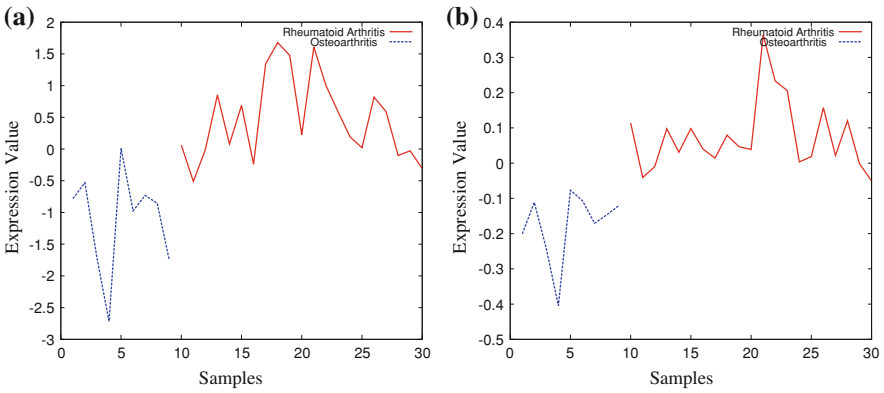

Fig. 9.7 Results obtained for RAOA data considering $\delta = 0.96$ **a** Initial representative **b** Augmented representative

data for $c = 3$, the actual gene performs better than augmented one with respect to the class separability index.

9.4.6 Performance of Coarse and Finer Clusters

In the MISAC algorithm, the augmented cluster representative is computed by averaging the genes of finer cluster, rather than all genes of corresponding coarse cluster. That is, instead of averaging all genes of coarse cluster, the augmented gene is computed by considering a subset of genes of coarse cluster, which is termed as the finer cluster, those increase the relevance value of initial cluster representative.

Table 9.3 Comparative performance analysis of supervised and unsupervised similarity measures for different data sets

Value of c	Measure	Breast		Leukemia		Colon		RAHC		RAOA	
		Supervised	Unsupervised	Supervised	Unsupervised	Supervised	Unsupervised	Supervised	Unsupervised	Supervised	Unsupervised
1	SVM	100	89.8	98.6	90.3	100	88.7	100	94.0	100	96.7
	K-NN	100	87.8	100	93.1	100	87.1	100	100	100	96.7
	NB	100	89.8	100	94.4	100	88.7	100	100	100	100
	CS	0.53	1.09	0.49	1.00	0.82	1.33	0.70	0.72	0.70	1.14
2	SVM	100	87.8	100	94.4	100	83.9	100	100	100	93.3
	K-NN	100	85.7	100	94.4	100	85.5	100	100	100	93.3
	NB	100	89.8	100	98.6	100	88.7	100	100	100	100
	CS	0.50	1.65	0.98	0.77	0.94	1.56	0.87	0.66	0.99	1.09
3	SVM	100	91.8	100	93.1	100	87.1	100	98.0	100	93.3
	K-NN	100	91.8	100	94.4	100	87.1	100	98.0	100	96.7
	NB	100	95.9	100	98.6	100	88.7	100	98.0	100	96.7
	CS	0.61	1.51	1.12	0.80	0.97	1.84	0.88	0.76	1.96	1.07

Table 9.4 Comparative performance analysis of augmented and actual genes for different data sets

Value of c	Measure	Breast		Leukemia		Colon		RAHC		RAOA	
		Augmented	Actual	Augmented	Actual	Augmented	Actual	Augmented	Actual	Augmented	Actual
1	SVM	100	85.7	98.6	90.3	100	83.9	100	88.0	100	93.3
	K-NN	100	89.8	100	93.1	100	83.9	100	88.0	100	90.0
	NB	100	89.8	100	94.4	100	83.9	100	68.0	100	93.3
	CS	0.53	1.16	0.49	1.00	0.82	1.83	0.70	4.38	0.70	0.87
2	SVM	100	85.7	100	94.4	100	83.9	100	72.0	100	90.0
	K-NN	100	91.8	100	94.4	100	83.9	100	84.0	100	86.7
	NB	100	89.8	100	98.6	100	88.7	100	88.0	100	86.7
	CS	0.50	1.56	0.98	0.77	0.94	1.86	0.87	7.93	0.99	1.07
3	SVM	100	87.8	100	93.1	100	80.6	100	84.0	100	93.3
	K-NN	100	89.8	100	94.4	100	85.5	100	86.0	100	90.0
	NB	100	91.8	100	98.6	100	83.9	100	94.0	100	83.3
	CS	0.61	1.97	1.12	0.80	0.97	2.31	0.88	4.03	1.96	1.27

Table 9.5 Comparative performance analysis of means of coarse and finer clusters

Value of c	Measure	Breast		Leukemia		Colon		RAHC		RAOA	
		Finer	Coarse	Finer	Coarse	Finer	Coarse	Finer	Coarse	Finer	Coarse
1	SVM	100	63.3	98.6	66.7	100	64.5	100	70.0	100	70.0
	K-NN	100	69.4	100	66.7	100	62.9	100	64.0	100	66.7
	NB	100	63.3	100	73.6	100	61.3	100	60.0	100	80.0
	CS	0.53	5.38	0.49	7.69	0.82	68.47	0.70	11.14	0.70	119.53
2	SVM	100	59.2	100	72.2	100	64.5	100	70.0	100	70.0
	K-NN	100	55.1	100	68.1	100	61.3	100	58.0	100	66.7
	NB	100	59.2	100	68.1	100	61.3	100	66.0	100	80.0
	CS	0.50	7.63	0.98	13.66	0.94	70.62	0.87	12.74	0.99	118.23
3	SVM	100	59.2	100	73.6	100	64.5	100	82.0	100	70.0
	K-NN	100	55.1	100	65.3	100	61.3	100	82.0	100	66.7
	NB	100	67.4	100	75.0	100	61.3	100	62.0	100	80.0
	CS	0.61	8.85	1.12	20.95	0.97	72.95	0.88	12.62	1.96	138.51

Table 9.5 presents the comparative performance of means computed from coarse cluster and that from finer cluster. The comparison is reported for $c = 3$ with respect to the class separability index and classification accuracy of the SVM, K-NN rule, and NB classifier. The results reported in Table 9.5 establish the fact that the augmented cluster representative obtained from finer cluster performs significantly better than that of coarse cluster, irrespective of the data sets and quantitative indices used. The attributes those present in the coarse cluster, but not in the corresponding finer cluster, are not responsible to increase the relevance value with respect to the class labels or response variables. Also, they degrade the quality of solution. Hence, the augmented cluster representatives should be computed by considering only genes of finer clusters, not all genes of coarse clusters.

9.4.7 Comparative Performance Analysis

Finally, Table 9.6 compares the best performance of the MISAC algorithm [31] with that of some existing algorithms such as ACA [2], MBBC [22], SGCA [5], GS [16], and mRMR [8]. The results are presented based on the best classification accuracy of the SVM, K-NN rule, and NB classifier for five microarray data sets. The values of δ for the MISAC algorithm are considered as 0.97, 0.96, 0.93, 0.98, and 0.96 for breast cancer, leukemia, colon cancer, RAHC, and RAOA data sets, respectively. From the results reported in Table 9.6, it is seen that the MISAC algorithm generates a set of clusters having highest classification accuracy of the SVM, K-NN rule, and NB classifier, and lowest class separability index values for all the cases. The better performance of the MISAC algorithm is achieved due to the fact that it can identify

Table 9.6 Comparative performance analysis of different methods on five microarray data sets

Data sets	Methods / Algorithms	$c = 1$				$c = 2$				$c = 3$			
		SVM	K-NN	NB	CS	SVM	K-NN	NB	CS	SVM	K-NN	NB	CS
Breast	MISAC	100	100	100	0.53	100	100	100	0.50	100	100	100	0.61
	ACA	81.6	81.6	81.6	2.07	81.6	83.7	81.6	2.92	89.8	83.7	83.7	1.03
	MBBC	79.6	79.6	85.7	1.04	79.6	81.6	85.7	1.18	81.6	81.6	89.8	0.94
	SGCA	100	100	100	1.74	100	100	100	1.29	100	100	100	1.83
	GS	75.5	79.6	79.6	1.68	85.7	85.7	83.7	2.60	89.8	87.8	85.7	3.75
	mRMR	85.7	89.8	89.8	1.16	81.6	89.8	95.9	1.70	93.9	95.9	100	1.54
Leukemia	MISAC	98.6	100	100	0.49	100	100	100	0.98	100	100	100	1.12
	ACA	82.4	82.4	88.2	1.69	88.2	82.4	88.2	1.19	88.2	91.2	88.2	3.25
	MBBC	88.2	88.2	90.3	0.94	94.4	91.7	93.1	1.89	94.4	93.1	95.8	1.63
	SGCA	93.1	94.4	94.4	1.16	94.4	94.4	94.4	2.01	94.4	95.8	94.4	1.76
	GS	97.2	94.4	91.7	0.43	97.2	97.2	95.8	1.67	100	100	94.4	1.90
	mRMR	90.3	93.1	94.4	1.00	94.4	94.4	98.6	1.46	94.4	95.8	100	1.08
Colon	MISAC	100	100	100	0.82	100	100	100	0.94	100	100	100	0.97
	ACA	72.6	77.4	64.5	3.08	72.6	83.9	75.8	1.46	77.4	83.9	64.5	2.59
	MBBC	64.5	64.5	72.6	1.68	75.8	72.6	72.6	1.69	75.8	75.8	82.3	3.05
	SGCA	72.6	72.6	75.8	5.10	75.8	77.4	77.4	3.80	77.4	77.4	77.4	4.25
	GS	83.9	82.3	82.3	1.41	82.3	83.9	79.0	2.70	87.1	87.1	85.5	4.10
	mRMR	83.9	83.9	83.9	1.83	83.9	83.9	83.9	2.51	75.8	83.9	83.9	3.89
RAHC	MISAC	100	100	100	0.70	100	100	100	0.87	100	100	100	0.88
	ACA	90.0	88.0	88.0	2.79	90.0	96.0	92.0	4.81	92.0	92.0	92.0	3.02
	MBBC	86.0	84.0	84.0	1.15	84.0	88.0	90.0	2.09	90.0	92.0	90.0	1.77
	SGCA	92.0	92.0	92.0	1.76	90.0	96.0	92.0	3.08	90.0	96.0	92.0	2.17
	GS	64.0	68.0	62.0	5.26	84.0	82.0	72.0	8.78	88.0	88.0	66.0	13.23
	mRMR	88.0	88.0	68.0	4.38	84.0	90.0	96.0	3.57	92.0	90.0	98.0	3.35
RAOA	MISAC	100	100	100	0.70	100	100	100	0.99	100	100	100	1.96
	ACA	86.7	83.3	83.3	1.90	86.7	83.3	83.3	2.11	86.7	86.7	86.7	1.54
	MBBC	86.7	86.7	83.3	1.91	86.7	86.7	90.0	2.06	86.7	86.7	86.7	3.88
	SGCA	93.3	90.0	90.0	1.71	93.3	93.3	96.7	3.04	93.3	93.3	96.7	1.67
	GS	80.0	73.3	83.3	1.46	93.3	96.7	80.0	2.94	86.7	93.3	83.3	4.61
	mRMR	93.3	90.0	93.3	0.87	96.7	96.7	90.0	1.26	96.7	100	90.0	2.01

functional groups of genes present in the microarray data sets more accurately than the existing algorithms. However, with respect to the class separability index, mRMR [8] for $c = 3$ and GS [16] for $c = 1$ perform better than the MISAC algorithm in case of leukemia data and for RAOA data at $c = 3$, both ACA [2] and SGCA [5] attain lower class separability index values than the MISAC algorithm. In this regard, it should be noted that the method proposed by Golub et al. [13] achieves maximum accuracy of 98.0, 98.6, 91.9, 98.0, and 96.7 % for breast, leukemia, colon, RAHC, and RAOA data sets, respectively.

Table 9.7 Significant shared GO terms for genes in best clusters obtained by different methods

Data sets	Methods / Algorithms	Biological process			Molecular function		Cellular component	
		Gene ontology term	*p*-value	FDR	*p*-value	FDR	*p*-value	FDR
Breast	MISAC	Positive regulation of biological process	5.7E-027	0	6.9E-024	0	2.1E-020	0
	GS	*	*	*	2.7E-03	4	*	*
	SGCA	*	*	*	*	*	*	*
Leukemia	MISAC	Multicellular organismal development	2.1E-02	10	2.4E-03	2	2.1E-03	0
	GS	*	*	*	2.3E-03	2	1.9E-03	4
	SGCA	*	*	*	4.8E-03	0	*	*
Colon	MISAC	Cellular process	1.8E-012	0	9.0E-016	0	2.0E-09	0
	GS	Regulation of system process	1.2E-02	32	4.5E-03	0	1.9E-03	0
	SGCA	Blood vessel development	1.7E-02	20	*	*	*	*
RAOA	MISAC	Immune system process	8.3E-07	0	2.5E-03	2	*	*
	GS	Immune system process	2.9E-016	0	9.9E-04	0	7.9E-04	0
	SGCA	Interspecies interaction between organisms	7.4E-03	18	3.0E-03	2	*	*

9.4.8 Biological Significance Analysis

To interpret the biological significance of the generated clusters, the gene ontology (GO) Term Finder is used [4]. As described in Chap. 8, the GO Term Finder finds the most significantly enriched GO terms associated with the genes belonging to a cluster. It determines whether any GO term annotates a specified list of genes at a frequency greater than that would be expected by chance, calculating the associated p-value by using the hypergeometric distribution and the Bonferroni multiple-hypothesis correction [4]. The closer the p-value is to zero, the more significant the particular GO term associated with the group of genes is, that is, the less likely the observed annotation of the particular GO term to a group of genes occurs by chance. On the other hand, the false discovery rate (FDR) is a multiple-hypothesis testing error measure indicating the expected proportion of false positives among the set of significant results. The FDR is particularly useful in the analysis of high-throughput data such as microarray gene expression.

Hence, the GO Term Finder is used to determine the statistical significance of the association of a particular GO term with the genes of best cluster produced by the MISAC algorithm. The GO Term Finder is used to compute both p-value and FDR (%) for all the GO terms from the biological processes (BP), molecular functions (MF), and cellular components (CC) ontology, and the most significant term, that is, the one with the lowest p-value, is chosen to represent the set of genes of best cluster. Table 9.7 presents the significant shared GO terms for the BP, along with the p-values and FDR for the BP, MF, and CC on different data sets. The results corresponding to the best clusters of some existing algorithms such as GS [16] and SGCA [5] are also provided on same data sets for the sake of comparison. The '*' in Table 9.7 represents that no significant shared term is found considering p-value cutoff as 0.05. From the results reported in Table 9.7, it is seen that the best cluster generated by the MISAC algorithm can be assigned to the GO biological processes with high reliability in terms of p-value and FDR. That is, the MISAC algorithm describes accurately the known classification, the one given by the GO, and thus reliable for extracting new biological insights.

9.5 Conclusion and Discussion

The problem of supervised gene clustering is addressed in this chapter. After explaining the merits and demerits of unsupervised gene clustering and existing supervised attribute clustering algorithms for microarray sample classification, a new mutual information-based supervised attribute clustering (MISAC) algorithm is presented to find co-regulated clusters of genes whose collective expression is strongly associated with the sample categories. It is based on a new quantitative measure, which incorporates the information of sample categories or class labels to calculate the similarity between two genes. Finally, the performance of the MISAC algorithm and

some existing methods is compared using the class separability index and predictive accuracy of support vector machine, K-nearest neighbor rule, and naive Bayes classifier.

For five microarray data, significantly better results are found for the MISAC algorithm compared to existing methods, irrespective of the classifiers used. All the results reported in this chapter demonstrate the feasibility and effectiveness of the MISAC algorithm. It is capable of identifying co-regulated clusters of genes whose average expression is strongly associated with the sample categories. The identified gene clusters may contribute to revealing underlying class structures, providing a useful tool for the exploratory analysis of biological data. Recently, a fuzzy-rough supervised attribute clustering algorithm [30] and a mutual information-based supervised gene clustering algorithm [32] have been reported to reveal various groups of co-regulated genes with strong association to the response variables.

While Chaps. 8 and 9 address the problems of unsupervised and supervised gene clustering, respectively, Chap. 10 discusses another important problem of microarray gene expression data sets, called biclustering.

References

1. Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc Nat Acad Sci USA 96(12):6745–6750
2. Au WH, Chan KCC, Wong AKC, Wang Y (2005) Attribute clustering for grouping, selection, and classification of gene expression data. IEEE/ACM Trans Comput Biol Bioinf 2(2):83–101
3. Battiti R (1994) Using mutual information for selecting features in supervised neural net learning. IEEE Trans Neural Networks 5(4):537–550
4. Boyle EI, Weng S, Gollub J, Jin H, Botstein D, Cherry JM, Sherlock G (2004) GO: term finder open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. Bioinformatics 20(18):3710–3715
5. Dettling M, Buhlmann P (2002) Supervised clustering of genes. Genome Biol 3(12):1–15
6. Devijver PA, Kittler J (1982) Pattern recognition: a statistical approach. Prentice Hall, Englewood Cliffs
7. Dhillon I, Mallela S, Kumar R (2003) Divisive information-theoretic feature clustering algorithm for text classification. J Mach Learn Res 3:1265–1287
8. Ding C, Peng H (2005) Minimum redundancy feature selection from microarray gene expression data. J Bioinf Comput Biol 3(2):185–205
9. Domany E (2003) Cluster analysis of gene expression data. J Stat Phys 110(3–6):1117–1139
10. Duda RO, Hart PE, Stork DG (1999) Pattern Classification and Scene Analysis. John Wiley and Sons, New York
11. Eisen MB, Spellman PT, Brown PO, Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. Proc Nat Acad Sci USA 95(25):14863–14868
12. Fukunaga K (1990) Introduction to statistical pattern recognition. Academic Press, New York
13. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, Lander ES (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286(5439):531–537
14. Haiying W, Huiru Z, Francisco A (2007) Poisson-based self-organizing feature maps and hierarchical clustering for serial analysis of gene expression data. IEEE/ACM Trans Comput Biol Bioinf 4(2):163–175

15. Hastie T, Tibshirani R, Botstein D, Brown P (2001) Supervised harvesting of expression trees. Genome Biol 1:1–12
16. Hastie T, Tibshirani R, Eisen MB, Alizadeh A, Levy R, Staudt L, Chan WC, Botstein D, Brown P (2000) 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. Genome Biol 1(2):1–21
17. Herrero J, Valencia A, Dopazo J (2001) A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics 17:126–136
18. Heyer LJ, Kruglyak S, Yooseph S (1999) Exploring expression data: identification and analysis of coexpressed genes. Genome Res 9(11):1106–1115
19. Huang D, Chow TWS (2004) Effective feature selection scheme using mutual information. Neurocomputing 63:325–343
20. Jain AK, Dubes RC (1988) Algorithms for clustering data. Prentice Hall, Englewood Cliffs, NJ
21. Jiang D, Tang C, Zhang A (2004) Cluster analysis for gene expression data: a survey. IEEE Trans Knowl Data Eng 16(11):1370–1386
22. Joo Y, Booth JG, Namkoong Y, Casella G (2008) Model-based bayesian clustering (MBBC). Bioinformatics 24(6):874–875
23. Jornsten R, Yu B (2003) Simultaneous gene clustering and subset selection for sample classification via MDL. Bioinformatics 19(9):1100–1109
24. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97(1–2):273–324
25. Koller D, Sahami M (1996) Toward optimal feature selection. In: Proceedings of the international conference on machine learning, pp 284–292
26. Li J, Su H, Chen H, Futscher BW (2007) Optimal search-based gene subset selection for gene array cancer classification. IEEE Trans Inf Technol Biomed 11(4):398–405
27. Liao JG, Chin KV (2007) Logistic regression for disease classification using microarray data: model selection in a large p and small n case. Bioinformatics 23(15):1945–1951
28. Liu X, Krishnan A, Mondry A (2005) An entropy based gene selection method for cancer classification using microarray data. BMC Bioinformatics 6(1):76
29. Maji P (2009) f-Information measures for efficient selection of discriminative genes from microarray data. IEEE Trans Biomed Eng 56(4):1063–1069
30. Maji P (2011) Fuzzy-rough supervised attribute clustering algorithm and classification of microarray data. IEEE Trans Syst Man Cybern Part B Cybern 41(1):222–233
31. Maji P (2012) Mutual information based supervised attribute clustering for microarray sample classification. IEEE Trans Knowl Data Eng 24(1):127–140
32. Maji P, Das C (2012) Relevant and Significant Supervised Gene Clusters for Microarray Cancer Classification. IEEE Transactions on NanoBioscience 11(2):161–168
33. McLachlan GJ, Do KA, Ambroise C (2004) Analyzing microarray gene expression data. John Wiley, Hoboken, NJ
34. Medvedovic M, Sivaganesan S (2002) Bayesian infinite mixture model based clustering of gene expression profiles. Bioinformatics 18(9):1194–1206
35. Nguyen D, Rocke D (2002) Tumor classification by partial least squares using microarray gene expression data. Bioinformatics 18:39–50
36. Peng H, Long F, Ding C (2005) Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 27(8):1226–1238
37. van der Pouw Kraan TCTM, van Gaalen FA, Kasperkovitz PV, Verbeet NL, Smeets TJM, Kraan MC, Fero M, Tak PP, Huizinga TWJ, Pieterman E, Breedveld FC, Alizadeh AA, Verweij CL (2003) Rheumatoid arthritis is a heterogeneous disease: evidence for differences in the activation of the STAT-1 pathway between rheumatoid tissues. Arthritis Rheum 48(8):2132–2145
38. van der Pouw Kraan TCTM, Wijbrandts CA, van Baarsen LGM, Voskuyl AE, Rustenburg F, Baggen JM, Ibrahim SM, Fero M, Dijkmans BAC, Tak PP, Verweij CL (2007) Rheumatoid arthritis subtypes identified by genomic profiling of peripheral blood sells: assignment of a type I interferon signature in a subpopulation of pateints. Ann Rheum Dis 66:1008–1014

39. Shannon C, Weaver W (1964) The mathematical theory of communication. University of Illinois Press, Champaign, IL
40. Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander ES, Golub TR (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. Proc Nat Acad Sci USA 96(6):2907–2912
41. Thalamuthu A, Mukhopadhyay I, Zheng X, Tseng GC (2006) Evaluation and comparison of gene clustering methods in microarray analysis. Bioinformatics 22(19):2405–2412
42. Vapnik V (1995) The nature of statistical learning theory. Springer-Verlag, New York
43. Wang L, Chu F, Xie W (2007) Accurate cancer classification using expressions of very few genes. IEEE/ACM Trans Comput Biol Bioinf 4(1):40–53
44. West M, Blanchette C, Dressman H, Huang E, Ishida S, Spang R, Zuzan H, Olson JA, Marks JR, Nevins JR (2001) Predicting the clinical status of human breast cancer by using gene expression profiles. Proc Nat Acad Sci USA 98(20):11462–11467
45. Yeung KY, Ruzzo WL (2001) Principal component analysis for clustering gene expression data. Bioinformatics 17(9):763–774

Chapter 10
Possibilistic Biclustering for Discovering Value-Coherent Overlapping δ-Biclusters

10.1 Introduction

The advent of DNA microarray technologies has revolutionized the experimental study of gene expression. Microarrays have been used to study different kinds of biological processes. It enables monitoring the transcription levels of many thousands of genes, while the cell undergoes specific conditions or processes [20]. The applications of such technology range from gene functional annotation and genetic network reconstruction to diagnosis of disease conditions and characterizing effects of medical treatment.

Clustering is one of the most popular approaches for analyzing gene expression data [15, 25] and has proved to be successful in many applications such as discovering gene pathway, gene classification, and function prediction [15, 20, 25]. Traditional clustering methods such as hierarchical clustering [23], k-means algorithm [22], and self organizing map [40] assume that genes in a cluster behave similarly over all the conditions. These methods produce reliable results for microarray experiments performed on homogeneous conditions. However, when the conditions of an experiment vary greatly, the assumption is no longer appropriate. In this regard, it is desirable to develop approaches that can detect those relevant conditions under which the behavior similarity between genes of a potential group exists. This leads to a promising paradigm of clustering, called biclustering.

The term biclustering [21] refers to a distinct class of clustering algorithms that performs simultaneous row–column clustering. The difference between clustering and biclustering is that clustering can be applied to either rows or columns of the data matrix separately. Biclustering, on the other hand, performs clustering in these two dimensions simultaneously. This means that clustering derives a global model while biclustering produces a local model. When clustering algorithms are used, each gene in a given gene cluster is defined using all conditions. Similarly, each condition in a condition cluster is characterized by the activity of all the genes that belong to it. However, each gene in a bicluster is selected using only a subset of the conditions and each condition in a bicluster is selected using only a subset of the genes. The

P. Maji and S. Paul, *Scalable Pattern Recognition Algorithms*,
DOI: 10.1007/978-3-319-05630-2_10,

goal of biclustering techniques is thus to identify subgroups of genes and subgroups of conditions, by performing simultaneous clustering of both rows and columns of gene expression matrix, instead of clustering these two dimensions separately.

Hence, unlike clustering algorithms, biclustering algorithms identify groups of genes that show similar activity patterns under a specific subset of the experimental conditions. Therefore, biclustering approach is the key technique to use when one or more of the following situations applies: (i) only a small set of the genes participates in a cellular process of interest; (ii) an interesting cellular process is active only in a subset of conditions; and (iii) a single gene may participate in multiple pathways that may or may not be coactive under all conditions. For these reasons, biclustering should identify groups of genes and conditions, obeying the following restrictions: (i) a cluster of genes should be defined with respect to only a subset of conditions; (ii) a cluster of conditions should be defined with respect to only a subset of genes; and (iii) the clusters should not be exclusive and/or exhaustive: a gene or condition should be able to belong to more than one cluster or to no cluster at all and be grouped using a subset of conditions or genes. Additionally, robustness in biclustering algorithms is especially relevant because of two additional characteristics of the systems under study. The first characteristic is the sheer complexity of gene regulation processes that require powerful analysis tools. The second characteristic is the level of noise in actual gene expression experiments that makes the use of intelligent statistical tools indispensable.

The term biclustering was initially introduced by Hartigan [21] as a way to cluster rows and columns of a matrix simultaneously for finding biclusters with minimum row variance. Based on the row variance, Tibshirani et al. [43] proposed a permutation-based method to induce the optimal number of biclusters. In addition to use the row variance defined by Hartigan [21], Cho et al. [12] also used the total squared residue to quantify the homogeneity of a bicluster. Therefore, their framework is applicable for finding both constant biclusters and value-coherent biclusters. However, the term biclustering was first used by Cheng and Church in gene expression data analysis [11] and an additive biclustering model was proposed in [11], for gene expression data by introducing the residue of an element in the bicluster and the mean squared residue of a submatrix. In addition, this method adjusts that measure to reject trivial biclusters by means of row variance. Yang et al. [51] generalized this additive biclustering model to incorporate null values and proposed a probabilistic algorithm, called flexible overlapped biclustering algorithm, that can discover a set of possibly overlapping biclusters simultaneously. Getz et al. [19] presented a coupled two-way clustering algorithm that uses hierarchical clustering applied separately to each dimension and then combines both results to get final biclusters. Tang et al. [42] presented an interrelated two-way clustering algorithm that combines the results of one-way clustering on both gene and sample dimensions to produce biclusters. Obviously, the quality of biclusters produced by these methods depends on the clusters generated at each dimension.

To discover row-constant or column-constant biclusters hidden in noisy data, several approaches have been proposed such as the model proposed by Califano et al. [7], Bayesian-based approach of Sheng et al. [37], probabilistic model of Segal et al.

[36], Gibbs sampling-based scheme of Wu et al. [49], plaid model of Lazzeroni and Owen [28], order preserving submatrix model of Ben-Dor et al. [1] and Liu and Wang [30], model of Murali and Kasif [33], and bipartite graph-based model of Tanay et al. [41]. To address the biclustering problem, recently several stochastic search techniques such as simulated annealing [6], evolutionary algorithm [14], and genetic algorithm [9] have also been employed. In [14], Divina and Aguilar-Ruiz proposed a biclustering method based on evolutionary algorithms that searches for biclusters following a sequential covering strategy. The main objective of this approach is to find biclusters of maximum size with mean squared residue lower than a given threshold. It also looks for biclusters with a relatively high row variance and with a low level of overlapping among biclusters. Lee et al. [29] and Sill et al. [38] used sparse singular value decomposition method to address the biclustering problem, while Sutheeworapong et al. [39] proposed a biclustering approach to analyze gene expression data with iterative optimization technique. Some other notable works are reported in [10, 35]. A survey on different biclustering algorithms for biological data analysis is reported in [31]. Also, the comparative analysis of different biclustering algorithms can be found in [17].

However, most of the algorithms reported earlier find exclusive biclusters, which is inappropriate in the biological context. Since biological processes are not independent of each other, many genes participate in multiple different processes. Each gene, therefore, should be assigned to multiple biclusters whenever biclusters are identified with processes. Hence, one of the main problems with biclustering technique is the uncertainty. Some of the sources of this uncertainty include incompleteness and vagueness in bicluster definitions of microarray data. In this background, two major soft computing techniques, namely, fuzzy sets theory [52] and rough sets theory [34], have gained popularity in modeling and propagating uncertainty. Recently, using rough sets and fuzzy sets, some biclustering algorithms have been proposed to discover value-coherent overlapping biclusters [8, 18, 44–48]. In these methods, the memberships of an entire row or column in different biclusters are computed directly to find out overlapping biclusters without considering the membership of each element or point of the gene expression data. The membership of an element is obtained either by multiplying or averaging the memberships of corresponding row and column of that element. In effect, there is no direct relation between the membership of an element and its residue value. However, the membership of each element must be dependent on its residue value directly to generate highly coherent biclusters.

In this chapter, a novel possibilistic biclustering (PBC) algorithm [13] is presented. The PBC is based on the concept of possibilistic clustering algorithm of Krishnapuram and Keller [27]. The PBC algorithm employs an iterative process to find biclusters of larger volume with stronger coherence and particularly with a high degree of overlapping. During each iteration of the PBC algorithm, the possibilistic memberships of all elements or points with respect to every bicluster are calculated and finally, the membership of an entire row or column is calculated from the memberships of the elements present in that row or column. Depending on that membership value, the biclusters are enlarged or degenerated. The main difference

between the PBC method and other existing overlapping methods is that the PBC method offers an efficient way to calculate the membership of every point rather than the membership of an entire row or column, which is desirable to generate value-coherent biclusters. Also, instead of taking contribution of all rows and columns, the PBC method considers only those rows and columns whose membership values are greater than a given threshold. Based on this criterion, a novel possibilistic biclustering algorithm is presented in this chapter to discover a set of overlapping biclusters with mean squared residue lesser than a predefined threshold δ. A mathematical analysis on the convergence property of the PBC algorithm is also reported. Some quantitative measures are presented for evaluating the quality of selected overlapping biclusters. The effectiveness of the PBC algorithm, along with a comparison with the existing algorithms, is demonstrated on yeast microarray data.

The structure of the rest of this chapter is as follows: Sect. 10.2 briefly introduces necessary definitions related to biclustering method. In Sect. 10.3, a new possibilistic biclustering algorithm, termed as the PBC, is presented. Some quantitative performance measures are introduced in Sect. 10.4, along with some existing measures, to evaluate the quality of selected biclusters. In Sect. 10.5, extensive experimental results of the PBC algorithm are discussed and compared to those generated by the algorithms of Cheng and Church [11], Divina and Aguilar-Ruiz [14], and Yang et el. [51]. Concluding remarks are given in Sect. 10.6.

10.2 Biclustering and Possibilistic Clustering

This section presents a brief introduction to the basic notions of possibilistic clustering and biclustering method.

10.2.1 Basics of Biclustering

Let $G = \{g_1, \cdots, g_i, \cdots, g_M\}$ and $E = \{c_1, \cdots, c_j, \cdots, c_N\}$ be the set of genes and set of experimental conditions involved in gene expression data measurement, respectively. The result can be represented by a matrix D with the set of rows G and set of columns E. Each element $a_{ij} \in D$ corresponds to the expression information of gene g_i in experiment c_j. A bicluster of a gene expression data is defined to be a subset of genes that exhibit similar behavior under a subset of experimental conditions, and vice versa. Thus, in the gene expression data matrix D, a bicluster will appear as a submatrix of D. This submatrix is denoted by the pair (I, J), where $|I| \leq |G|$ and $|J| \leq |E|$. The volume of a bicluster is defined as the number of elements a_{ij} such that $i \in I$ and $j \in J$ that will appear as a submatrix of D.

Example 10.1 Suppose the expression matrix D consists of 10 genes and 8 conditions as shown in Fig. 10.1, where the rows of the matrix represent the genes and the columns represent the conditions. Then, a bicluster defined over the matrix D could be $(\{1, 3, 5\}, \{2, 4, 7\})$, thus consisting of genes $\{g_1, g_3, g_5\}$ and of conditions

9	**1**	10	**1**	5	**7**	3	8
0	2	3	9	5	3	2	5
3	**2**	6	**1**	8	**6**	1	2
3	8	3	4	1	5	3	1
4	**1**	3	**2**	2	**5**	2	2
5	7	2	4	2	6	7	2
11	3	2	7	3	8	4	3
4	2	12	5	7	0	4	6
4	2	6	4	2	7	8	2
3	7	3	7	5	2	4	6

Fig. 10.1 Expression matrix: row and column represent gene and condition, respectively

$\{c_2, c_4, c_7\}$. The volume of this bicluster is 9. In Fig. 10.1, the elements belonging to the bicluster are highlighted.

Definition 10.1 Let (I_k, J_k) be the kth bicluster B_k, then the base of a gene g_i is defined as [11]

$$a_{iJ_k} = \frac{1}{|J_k|} \sum_{j \in J_k} a_{ij}, \tag{10.1}$$

while the base of a condition c_j is defined as [11]

$$a_{I_k j} = \frac{1}{|I_k|} \sum_{i \in I_k} a_{ij}. \tag{10.2}$$

The base of the bicluster B_k is the mean of all entries contained in (I_k, J_k), that is,

$$a_{I_k J_k} = \frac{1}{|I_k|.|J_k|} \sum_{i \in I_k} \sum_{j \in J_k} a_{ij}. \tag{10.3}$$

Note that in the above definitions, a_{iJ_k} and $a_{I_k j}$ correspond to the means of the ith row and jth column of the bicluster (I_k, J_k), respectively.

Definition 10.2 The residue of an entry a_{ij} of a bicluster (I_k, J_k) is given by

$$r_{ij} = (a_{ij} - a_{iJ_k} - a_{I_k j} + a_{I_k J_k}). \tag{10.4}$$

In order to quantify the difference between the actual value and expected value of an entry predicted from the corresponding gene base, condition base, and bicluster base, the concept of residue is used. The residue is an indicator of the degree of coherence of an element with respect to the remaining ones in the bicluster, given the tendency of the relevant gene and the relevant condition. The lower the residue, the stronger the coherence.

Definition 10.3 The mean squared residue of a bicluster (I_k, J_k) is defined as follows [11]:

$$H_k = \frac{1}{|I_k|.|J_k|} \sum_{i \in I_k} \sum_{j \in J_k} r_{ij}^2. \tag{10.5}$$

The mean squared residue is the variance of the set of all elements in the bicluster, plus the mean row variance and the mean column variance. The lower the mean squared residue, the stronger the coherence exhibited by the bicluster, and the better the quality of the bicluster. If a bicluster has a mean squared residue lower than a given threshold δ, then the bicluster is termed as a δ-bicluster.

Definition 10.4 Let (I_k, J_k) be a bicluster, then the row variance of (I_k, J_k) is defined as follows:

$$\text{VAR}_{I_k J_k} = \frac{1}{|I_k|.|J_k|} \sum_{i \in I_k} \sum_{j \in J_k} (a_{ij} - a_{i J_k})^2. \tag{10.6}$$

The row variance may be referred to be relatively large to reject trivial bicluster. By using row variance as an accompanying score, one wants to guarantee that the bicluster captures genes exhibiting fluctuating yet coherent trends under some set of conditions.

10.2.2 Possibilistic Clustering

One of the most widely used prototype-based fuzzy partitional clustering algorithms is fuzzy c-means [4]. It offers the opportunity to deal with the data that belongs to more than one cluster at the same time. It assigns memberships to an object which are inversely related to the relative distance of the object to cluster prototypes. Also, it can deal with the uncertainties arising from overlapping cluster boundaries.

However, the fuzzy c-means may be inaccurate in a noisy environment [27] as the resulting membership values do not always correspond well to the degrees of belonging of the data. In real data analysis, noise and outliers are unavoidable. Hence, to reduce this weakness, and to produce memberships that have a good explanation of the degrees of belonging for the data, Krishnapuram and Keller [27] proposed the possibilistic c-means algorithm that uses a possibilistic type of membership function to describe the degree of belonging. It partitions a set of objects X into c clusters by minimizing the objective function

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n} (\nu_{ij})^m ||x_j - v_i||^2 + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} (1 - \nu_{ij})^m \tag{10.7}$$

where η_i represents the scale parameter, m is a weighting exponent called the fuzzifier, $\nu = [\nu_{ij}]_{c \times n}$ is the membership matrix, and the value of ν_{ij} depends only

on the similarity between the object x_j and the centroid v_i. The resulting partition of the data can be interpreted as a possibilistic partition, and the membership values may be interpreted as degrees of possibility of the objects belonging to the classes, that is, the compatibilities of the objects with the centroids. A brief description of both fuzzy c-means and possibilistic c-means algorithms is available in Chap 8.

10.3 Possibilistic Biclustering Algorithm

In this section, a new possibilistic biclustering (PBC) algorithm, proposed by Das and Maji [13], is presented, incorporating the concept of possibilistic clustering algorithm of Krishnapuram and Keller [27] into biclustering framework. The integration of the possibilistic membership of fuzzy sets and biclustering algorithm makes the PBC algorithm effective for generating value-coherent overlapping biclusters with mean squared residue lower than a predefined threshold.

10.3.1 Objective Function

The PBC algorithm partitions a set of elements or points $\{a_{ij}\}$ of a gene expression data D into K biclusters by minimizing the following objective function

$$\overline{J} = \sum_{k=1}^{K} \sum_{i \in I_k} \sum_{j \in J_k} \mathscr{U}_{kij}^{m} r_{kij}^{2} + \sum_{k=1}^{K} \mathscr{H}_k \sum_{i \in I_k} \sum_{j \in J_k} (1 - \mathscr{U}_{kij})^{m} \tag{10.8}$$

where $\mathscr{U}_{kij} \in [0, 1]$ represents the possibilistic membership of the element or point $a_{ij} \in D$ into the kth bicluster B_k represented as (I_k, J_k), $m \in [1, \infty)$ is the fuzzifier, and K is the total number of biclusters. The term r_{kij} is the residue of the element a_{ij} in the kth bicluster, which has the similar expression as that in (10.4) and is given by

$$r_{kij} = (a_{ij} - a_{iJ_k} - a_{I_k j} + a_{I_k J_k}) \tag{10.9}$$

where a_{iJ_k}, $a_{I_k j}$, and $a_{I_k J_k}$ represent the means or bases of the ith gene, jth condition, and kth bicluster, respectively. Hence, the first term $(\mathscr{U}_{kij}^{m} r_{kij}^{2})$ of (10.8) is the fuzzy squared residue of the element a_{ij} in the kth bicluster. Note that the term $(1 - \mathscr{U}_{kij})^{m}$ in (10.8) is monotone decreasing function of $\mathscr{U}_{kij}$. This forces $\mathscr{U}_{kij}$ to be as large as possible to avoid the trivial solutions.

The parameter $\mathscr{H}_k$ determines the residue value for which the membership value of a point or element becomes 0.5. Hence, it needs to be chosen depending on the desired bandwidth of the possibility (membership) distribution for each bicluster. This value could be the same for all biclusters if all biclusters are expected to be similar. In general, it is desirable that $\mathscr{H}_k$ relates to the overall size and shape of

bicluster B_k. Also, it is to be noted that $\mathscr{H}_k$ determines the relative degree to which the second term in the objective function of (10.8) is important compared with the first. If the two terms are to be weighted roughly equally, then $\mathscr{H}_k$ should be of the order of r_{kij}^2. In this work, the following definition is used:

$$\mathscr{H}_k = P.\frac{\sum_{i\in I_k}\sum_{j\in J_k}\mathscr{U}_{kij}^m r_{kij}^2}{\sum_{i\in I_k}\sum_{j\in J_k}\mathscr{U}_{kij}^m}. \tag{10.10}$$

This choice makes $\mathscr{H}_k$ proportional to the fuzzy mean squared residue of bicluster B_k. The denominator of $\mathscr{H}_k$ represents the fuzzy cardinality of bicluster B_k. Typically, P is chosen to be 1.

Solving the objective function of (10.8) with respect to $\mathscr{U}_{kij}$, we get the possibilistic membership of an element $a_{ij} \in D$ into kth bicluster that follows next:

$$\mathscr{U}_{kij} = \frac{1}{1+\left(\frac{r_{kij}^2}{\mathscr{H}_k}\right)^{\frac{1}{m-1}}}. \tag{10.11}$$

Hence, in each iteration, the updated value of $\mathscr{U}_{kij}$ depends only on the residue r_{kij} of the element a_{ij} with respect to the kth bicluster B_k. If the value of r_{kij} is zero, then the membership of that point with respect to the kth bicluster is 1.0. If the value of r_{kij} is greater than zero, then the membership value of that point is less than 1.0 and 0.0 when $r_{kij} \to \infty$. Hence, $0.0 \leq \mathscr{U}_{kij} \leq 1.0$.

After computing the membership of each point $a_{ij} \in D$ with respect to all biclusters, the memberships of all rows and columns are computed with respect to all biclusters. If the kth bicluster B_k has $|I_k|$ rows and $|J_k|$ columns, then the membership of the ith row in the kth bicluster B_k is defined as follows:

$$\mathscr{U}_{kiJ} = \left[\frac{1}{|J_k|}\sum_{j\in J_k}\mathscr{U}_{kij}^m\right]^{\frac{1}{m}} \tag{10.12}$$

and the membership of the jth column in the kth bicluster is

$$\mathscr{U}_{kIj} = \left[\frac{1}{|I_k|}\sum_{i\in I_k}\mathscr{U}_{kij}^m\right]^{\frac{1}{m}}. \tag{10.13}$$

In each iteration, if the maximum membership value of the ith row (respectively, jth column) is greater than a threshold ξ and if the difference between the maximum

membership and the membership with respect to a particular bicluster of that row (respectively, column), which is greater than ξ, is less than a threshold η, then this row (respectively, column) will be selected for insertion into that particular bicluster. In this way, all rows and columns are examined with respect to all biclusters and are inserted successfully into the biclusters. Hence, the rows and columns, which are inserted in this process, have the membership values greater than ξ. Here, the threshold ξ is used to generate highly coherent biclusters. On the other hand, if the membership of a row or column is very high in some biclusters, then the row or column is inserted into only those biclusters. This is adjusted using the predefined threshold η.

After all insertion, the mean squared residue of each bicluster is computed and compared with a given threshold δ. If the mean squared residue value of the kth bicluster B_k is greater than δ, then the row or column with least membership value in that bicluster is deleted and the process is repeated until the value becomes less than δ. In this way, the PBC algorithm generate highly coherent overlapping biclusters with lower mean squared residue value.

10.3.2 Bicluster Means

In each iteration, the means or bases of the ith gene, jth condition, and kth bicluster are calculated to compute the mean squared residue of each bicluster based on the possibilistic membership values of all rows and columns in different biclusters. The means or bases of the ith gene, jth condition, and kth bicluster for the PBC algorithm are obtained by solving (10.8) with respect to a_{iJ_k}, a_{I_kj}, and $a_{I_kJ_k}$, respectively:

$$a_{iJ_k} = \frac{1}{|J_k|\mathcal{U}_{kiJ}^m}\left\{\sum_{j\in J_k}\mathcal{U}_{kij}^m(a_{ij} - a_{I_kj})\right\} + a_{I_kJ_k} \tag{10.14}$$

$$a_{I_kj} = \frac{1}{|I_k|\mathcal{U}_{kIj}^m}\left\{\sum_{i\in I_k}\mathcal{U}_{kij}^m(a_{ij} - a_{iJ_k})\right\} + a_{I_kJ_k} \tag{10.15}$$

$$a_{I_kJ_k} = \frac{\sum_{i\in I_k}\mathcal{U}_{kiJ}^m a_{iJ_k}}{\sum_{i\in I_k}\mathcal{U}_{kiJ}^m} + \frac{\sum_{j\in J_k}\mathcal{U}_{kIj}^m a_{I_kj}}{\sum_{j\in J_k}\mathcal{U}_{kIj}^m} - \frac{\sum_{i\in I_k}\sum_{j\in J_k}\mathcal{U}_{kij}^m a_{ij}}{\sum_{i\in I_k}\sum_{j\in J_k}\mathcal{U}_{kij}^m}. \tag{10.16}$$

Hence, the base of the ith gene a_{iJ_k} in the kth bicluster B_k depends on the means of all conditions present in that bicluster as well as the base of that bicluster. Similarly, the base of the jth condition a_{I_kj} in the B_k bicluster depends on the means of all genes present in the B_k and the base of B_k.

10.3.3 Convergence Condition

In this subsection, a mathematical analysis on the convergence property of the PBC algorithm is presented. In the PBC algorithm, the means or bases of the ith gene, jth condition, and kth bicluster are calculated using (10.14), (10.15), and (10.16), respectively. However, these three equations can be written as

$$\sum_{j \in J_k} \mathscr{U}_{kij}^m \cdot a_{iJ_k} = \sum_{j \in J_k} \mathscr{U}_{kij}^m (a_{ij} - a_{I_k j} + a_{I_k J_k}) \tag{10.17}$$

$$\sum_{i \in I_k} \mathscr{U}_{kij}^m \cdot a_{I_k j} = \sum_{i \in I_k} \mathscr{U}_{kij}^m (a_{ij} - a_{iJ_k} + a_{I_k J_k}) \tag{10.18}$$

$$\sum_{i \in I_k} \sum_{j \in J_k} \mathscr{U}_{kij}^m \cdot a_{I_k J_k} = \sum_{i \in I_k} \sum_{j \in J_k} \mathscr{U}_{kij}^m (a_{iJ_k} + a_{I_k j} - a_{ij}) \tag{10.19}$$

Hence, (10.17) represents a set of linear equations in terms of a_{iJ_k} if $\mathscr{U}_{kij}$, $a_{I_k j}$, and $a_{I_k J_k}$ are kept constant. Similarly, (10.18) and (10.19) represent a set of linear equations in terms of $a_{I_k j}$ and $a_{I_k J_k}$, respectively. A simple way to analyze the convergence property of the algorithm is to view (10.14), (10.15), and (10.16) as the Gauss-Seidel iterations for solving the set of linear equations. The Gauss-Seidel algorithm is guaranteed to converge if the matrix representing each equation is diagonally dominant [24]. This is a sufficient condition, not a necessary one. The iteration may or may not converge if the matrix is not diagonally dominant [24]. The matrix corresponding to (10.14) is given by:

$$\tilde{\mathrm{A}} = \begin{bmatrix} \tilde{a}_1 & 0 & \cdots & 0 \\ 0 & \tilde{a}_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \tilde{a}_{I_k} \end{bmatrix}; \quad \tilde{a}_i = \sum_{j \in J_k} \mathscr{U}_{kij}^m \tag{10.20}$$

Similarly, the matrices corresponding to (10.15) and (10.16) are given by

$$\tilde{\mathrm{B}} = \begin{bmatrix} \tilde{b}_1 & 0 & \cdots & 0 \\ 0 & \tilde{b}_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \tilde{b}_{J_k} \end{bmatrix}; \quad \tilde{b}_j = \sum_{i \in I_k} \mathscr{U}_{kij}^m \tag{10.21}$$

$$\tilde{\mathrm{C}} = \begin{bmatrix} \tilde{c}_1 & 0 & \cdots & 0 \\ 0 & \tilde{c}_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \tilde{c}_K \end{bmatrix}; \quad \tilde{c}_k = \sum_{i \in I_k} \sum_{j \in J_k} \mathscr{U}_{kij}^m \tag{10.22}$$

For $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ to be diagonally dominant, we must have

$$\tilde{a}_i \geq 0; \qquad \tilde{b}_j \geq 0; \qquad \tilde{c}_k \geq 0. \tag{10.23}$$

This is the sufficient condition for matrices $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ to be diagonally dominant. Under this condition, the iteration would converge if (10.14–10.16) are, repetitively, applied with $\mathscr{U}_{kij}$ kept constant. In practice, (10.11–10.13) and (10.14–10.16) are applied alternatively in the iterations. The condition in (10.23) is still correct according to Bezdek's convergence theorem of fuzzy c-means algorithm [2, 3] and Yan's convergence analysis of the fuzzy curve-tracing algorithm [50]. All three matrices $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ are also the Hessian (second order derivative) of $\overline{J}$ with respect to a_{iJ_k}, a_{I_kj}, and $a_{I_kJ_k}$. As all three matrices $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ are diagonally dominant, all their eigenvalues are positive. Also, the Hessian of $\overline{J}$ with respect to $\mathscr{U}_{kij}$ can be easily shown to be diagonal matrix and are positive definite. Thus, according to the theorem derived by Bezdek [2, 3] and the analysis done by Yan [50], it can be concluded that the PBC algorithm converges, at least along a subsequence, to a local optimum solution as long as the condition in (10.23) is satisfied. Intuitively, the objective function $\overline{J}$ reduces in all steps corresponding to (10.11–10.13) and (10.14–10.16), so the compound procedure makes the function $\overline{J}$ descent strictly.

10.3.4 Details of the Algorithm

Approximate optimization of $\overline{J}$ by the PBC algorithm is based on Picard iteration through (10.11–10.13) and (10.14–10.16). The process starts with a set of seeds or initial biclusters. The initial bases or means of a gene, condition, and bicluster are calculated using (10.1), (10.2), and (10.3), respectively. These bases serve as the initial solutions for the PBC algorithm. The possibilistic memberships of all elements or points are calculated using (10.11). The scale parameters $\mathscr{H}_k$ for K biclusters are obtained using (10.10), while (10.12) and (10.13) are used to compute the memberships of all rows and columns with respect to different biclusters, respectively. Let

$$\mathscr{U}_{iJ}^{\max} = \max_{1 \leq k \leq K} \{\mathscr{U}_{kiJ}\} \tag{10.24}$$

$$\mathscr{U}_{Ij}^{\max} = \max_{1 \leq k \leq K} \{\mathscr{U}_{kIj}\} \tag{10.25}$$

be the maximum memberships of the ith row and jth column, respectively. For any row i and for any bicluster B_k, if $\mathscr{U}_{iJ}^{\max} < \xi$, then the ith row will not be inserted into any bicluster. Otherwise, compute the member set $\mathscr{S}$, where

$$\mathscr{S} = \{k | \mathscr{U}_{kiJ} \geq \xi \text{ and } (\mathscr{U}_{iJ}^{\max} - \mathscr{U}_{kiJ}) \leq \eta : 1 \leq k \leq K\}. \tag{10.26}$$

After computing the set $\mathscr{S}$, the ith row is inserted into the kth bicluster present in the set $\mathscr{S}$. That is, the ith row is not inserted into any other biclusters that are not present in this set. Similar decision is taken for any column j. The parameters ξ and η are two predefined thresholds. After assigning each row and column in different biclusters, the new means of the genes, conditions, and biclusters are calculated as per (10.14), (10.15), and (10.16), respectively. Also, the mean squared residue of each bicluster is computed as per (10.5) and compared with a given threshold δ. If the mean squared residue value of the kth bicluster B_k is greater than δ, then the row or column with least membership value in that bicluster is deleted and the process is repeated until the value becomes less than δ. In this way, the PBC algorithm generates highly coherent overlapping biclusters with lower mean squared residue value. Based on the above discussions, the PBC algorithm is outlined next.

- **Input:** $\{B_k : 1 \leq k \leq K\}$, a set of K seeds or initial biclusters.
- **Output:** $\{\overline{B}_k : 1 \leq k \leq K\}$, a set of K overlapping biclusters.

1. For each object (gene or condition) **v**, do:
   a. Calculate the membership of each point of the object **v** with respect to every bicluster using (10.11).
   b. Calculate the membership of the object **v** with respect to every bicluster using (10.12) or (10.13) and find the maximum memberships using (10.24) or (10.25).
   c. If the maximum membership is less than ξ, then goto step 1.
   d. If the maximum membership is greater than ξ, then compute the set $\mathscr{S}$ using (10.26) and do:
   e. Insert object **v** into all those biclusters that are present in set $\mathscr{S}$.
2. For each bicluster B_k, do:
   a. Compute new means for genes, conditions, and bicluster using (10.14), (10.15), and (10.16), respectively.
   b. Calculate mean squared residue H_k of bicluster B_k as per (10.5).
   c. If $H_k > \delta$, then
      i. Compute the membership of every object.
      ii. Delete the object with minimum membership value and repeat this step until the condition becomes false.
3. Goto step 1 until the termination condition for adjustment is satisfied.
4. For each bicluster B_k, do:
   a. Calculate the row variance of every row.
   b. If the row variance of a particular row is 0, then delete this row that may lead to a constant bicluster.
5. Output the best solution.
6. End.

10.3.5 Termination Condition

The process iterates until the termination condition is met. In order to terminate the PBC algorithm, the overall mean squared residue per total volume is used that follows next:

$$Overall(H/V) = \frac{H}{V}; \tag{10.27}$$

where

$$V = \sum_{k=1}^{K} V_k = \sum_{k=1}^{K} |I_k|.|J_k|; \tag{10.28}$$

and

$$H = \sum_{k=1}^{K} H_k = \sum_{k=1}^{K} \frac{1}{|I_k|.|J_k|} \sum_{i \in I_k} \sum_{j \in J_k} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2. \tag{10.29}$$

Here H_k represents the mean squared residue of the kth bicluster B_k and V_k represents the volume of that bicluster. The value of *Overall(H/V)* reflects the overall quality of the biclusters generated by the PBC algorithm. The lower the *Overall(H/V)* value is, the higher the coherence of biclusters exhibits. The algorithm will terminate when the *Overall(H/V)* in last iteration is less than or equal to the *Overall(H/V)* in the current iteration.

10.3.6 Selection of Initial Biclusters

The PBC algorithm starts with a set of seeds as initial biclusters and carries out an iterative process to improve the overall quality of the biclusters. Intuitively, seeds that demonstrate higher coherence will facilitate refining biclusters with lesser iteration steps. This concern is addressed here within the framework of two-way clustering [19, 42].

In this work, the k-medoids algorithm [26] is used on the gene or row and condition or column dimensions of the expression data matrix separately and then the results are combined to obtain a set of seeds that are basically small coregulated submatrices. Given a gene expression data matrix D with M rows and N columns, let k_g be the number of clusters in gene dimension and k_c be the number of clusters in condition dimension after using k-medoids algorithm. Let C^g be the set of gene clusters and C^c denotes the set of condition clusters. Let $c_i^g \in C^g$ $(1 \leq i \leq k_g)$ and $c_j^c \in C^c$ $(1 \leq j \leq k_c)$. The pair (c_i^g, c_j^c) denotes a submatrix of D. Therefore, by combining the results of gene-dimensional k-medoids clustering and condition-dimensional k-medoids clustering, total $(k_g \times k_c)$ seeds are obtained.

The two-way clustering results exhibit similarity on either gene dimension or condition dimension. However, they may not well capture the overall coherence of both a subset of genes and a subset of conditions. To improve the quality of seeds obtained by two-way k-medoids algorithm, the single node deletion algorithm of Cheng and Church [11] is used. If the mean squared residue H_k of each bicluster (seed) B_k is greater than a predefined threshold δ, then remove the row or column whichever with the larger residue value and this step is repeated until the mean squared residue of each seed becomes less than or equal to the threshold δ. Among these refined seeds, the best $K \leq (k_g \times k_c)$ seeds those exhibit relatively higher coherence and larger size are chosen as inputs for the PBC algorithm. The algorithm to generate highly coherent seeds using k-medoids algorithm is described next.

- **Input:** Gene expression matrix D with M rows and N columns.
- **Output:** A set of K-initial biclusters.

1. Perform k-medoids clustering on gene dimension.
2. Perform k-medoids clustering on condition dimension.
3. Combine gene clusters and condition clusters to get K initial biclusters.
4. For each initial bicluster B_k, $(1 \leq k \leq K)$, do:
   a. Calculate the mean squared residue H_k of each bicluster B_k.
   b. If mean squared residue H_k of bicluster B_k is greater than the threshold δ, then delete row or column whichever with larger residue value and repeat this step until the mean squared residue $H_k \leq \delta$.
5. End.

10.4 Quantitative Indices

In this section, a new quantitative index $\mathscr{R}$ called degree of overlapping is presented, along with some existing indices, to evaluate quantitatively the quality of generated biclusters.

10.4.1 Average Number of Genes

The average number of genes I_{avg} is defined as follows:

$$I_{\text{avg}} = \frac{1}{K} \sum_{k=1}^{K} |I_k|, \tag{10.30}$$

where K is the total number of biclusters and $|I_k|$ represents the number of genes present in the kth bicluster.

10.4.2 Average Number of Conditions

The average number of conditions J_{avg} is defined as:

$$J_{\text{avg}} = \frac{1}{K} \sum_{k=1}^{K} |J_k|, \tag{10.31}$$

where $|J_k|$ represents the number of experiments or conditions present in the kth bicluster B_k.

10.4.3 Average Volume

The average volume V_{avg} can be defined as follows:

$$V_{\text{avg}} = \frac{1}{K} \sum_{k=1}^{K} V_k = \frac{1}{K} \sum_{k=1}^{K} |I_k|.|J_k|. \tag{10.32}$$

Here V_k represents the volume of the kth bicluster B_k, which is equal to the number of elements a_{ij} such that $i \in I_k$ and $j \in J_k$.

10.4.4 Average Mean Squared Residue

The average mean squared residue H_{avg} is defined as:

$$H_{\text{avg}} = \frac{1}{K} \sum_{k=1}^{K} H_k; \tag{10.33}$$

where

$$H_k = \frac{1}{|I_k|.|J_k|} \sum_{i \in I_k} \sum_{j \in J_k} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2. \tag{10.34}$$

Here H_k represents the mean squared residue of the kth bicluster B_k. The lower the value of H_{avg} is, the higher the coherence of biclusters exhibits and the better the quality of the generated bicluster.

10.4.5 Degree of Overlapping

The degree of overlapping $\mathscr{R}$ among all biclusters is defined as follows:

$$\mathscr{R} = \frac{1}{|G|.|E|} \sum_{i=1}^{|G|} \sum_{j=1}^{|E|} \Gamma_{ij} \tag{10.35}$$

$$\text{and } \Gamma_{ij} = \frac{1}{(K-1)} \left\{ \sum_{k=1}^{K} w_k(a_{ij}) - 1 \right\} \tag{10.36}$$

where K is the total number of biclusters, $|G|$ represents the total number of genes, and $|E|$ represents the total number of experiments or conditions in the expression data matrix D. The value of $w_k(a_{ij})$ is either 0 or 1. If the element or point $a_{ij} \in D$ is present in the kth bicluster, then $w_k(a_{ij}) = 1$, otherwise 0. Hence, the $\mathscr{R}$ index represents the degree of overlapping among the biclusters. As the degree of overlapping increases, the value of $\mathscr{R}$ index also increases. Therefore, for a given data matrix and δ value, the higher the $\mathscr{R}$ index value, the higher would be the degree of overlapping of the generated biclusters. Also, $0.0 \leq \mathscr{R} \leq 1.0$.

10.5 Experimental Results

In order to assess the performance of the PBC algorithm, the method is compared with the existing methods of Cheng and Church (henceforth termed as CC) [11], Yang et al. (henceforth termed as FLOC) [51], and Divina and Aguilar-Ruiz (henceforth termed as SEBI) [14]. The major metrics for evaluating the performance of different algorithms are the index $\mathscr{R}$ introduced in Sect. 10.4, as well as some existing measures such as I_{avg}, J_{avg}, V_{avg}, and H_{avg}.

The experiments are conducted on the well-known yeast microarray data set. It is the yeast *Saccharomyces cerevisiae* cell cycle expression data set, where the expression matrix consists of 2884 genes and 17 experimental conditions. This data set is taken from [11], where the original data is processed, replacing missing values with random values. The value of δ for the yeast data set is taken from [11] and the number of clusters for two-way k-medoids algorithm is set as the square root of the number of objects. In the current study, the parameters used are as follows: $\delta = 300$, $k_g = 54$, and $k_c = 5$. The PBC algorithm is implemented in C language and run in LINUX environment having machine configuration Pentium IV, 3.2 GHz, 1 MB cache and 1 GB RAM.

Table 10.1 Performance of the PBC for Different Values of m, ξ, and η

m	ξ	η	H_{avg}	V_{avg}	I_{avg}	J_{avg}	$\mathscr{R}$
1.2	0.55	0.15	180.10	1737.48	256.17	7.81	0.029
		0.20	185.07	2351.75	356.45	7.41	0.042
		0.25	187.84	2674.53	409.44	7.17	0.047
	0.60	0.15	172.24	1666.22	251.38	8.14	0.028
		0.20	180.66	2195.17	321.15	7.87	0.039
		0.25	183.67	2742.18	436.02	7.41	0.051
	0.65	0.15	167.36	1616.34	230.79	8.72	0.027
		0.20	154.12	1047.64	139.15	8.95	0.011
		0.25	164.74	1767.66	267.29	8.63	0.029
1.6	0.55	0.15	180.69	1889.13	276.21	7.41	0.032
		0.20	184.11	2104.63	308.15	7.56	0.039
		0.25	189.57	2187.09	319.25	7.53	0.041
	0.60	0.15	183.13	2019.30	294.80	7.72	0.035
		0.20	185.03	2117.16	387.12	7.72	0.039
		0.25	189.27	2089.56	392.05	7.74	0.038
	0.65	0.15	172.78	1629.83	234.94	8.01	0.026
		0.20	174.58	2138.95	320.66	8.28	0.038
		0.25	179.20	2037.59	322.52	8.08	0.036
2.0	0.55	0.15	191.83	1934.76	311.17	6.70	0.033
		0.20	197.01	2068.34	389.15	6.74	0.035
		0.25	199.67	2083.51	392.17	6.81	0.033
	0.60	0.15	181.89	1966.40	313.42	6.89	0.034
		0.20	187.08	2063.10	376.44	6.70	0.041
		0.25	194.51	2114.25	382.37	6.85	0.039
	0.65	0.15	179.82	1852.22	282.45	7.68	0.031
		0.20	181.52	2132.42	331.44	7.49	0.037
		0.25	183.17	2169.06	342.11	7.67	0.037

10.5.1 Optimum Values of Different Parameters

This section presents the performance of the PBC algorithm on yeast microarray gene expression data set for different parameter values. The fuzzifier m, and thresholds ξ and η play an important role in the PBC algorithm. The addition or deletion of a particular row or column is done based on the values of these two thresholds. The row and columns having membership values less than ξ are not considered for addition in any bicluster. If a bicluster contains rows and columns with membership values larger than ξ, then the bicluster will be highly coherent depending on the value of ξ. That is, the threshold ξ controls the degree of coherence of the bicluster. The higher the value of ξ, the higher would be the coherence of the biclusters. On the other hand, the contribution of each row or column in multiple biclusters is controlled by both parameters ξ and η. If the value of ξ is small and the value of η is high, then the degree of overlapping of the generated biclusters will be high with lesser degree

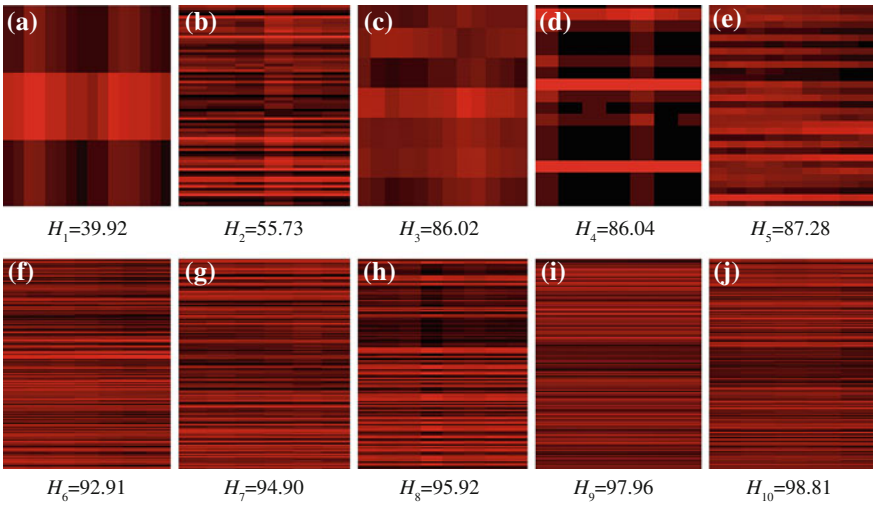

Fig. 10.2 Eisen plots of best ten biclusters and corresponding mean squared residue values obtained using the PBC algorithm

of coherence. Hence, both ξ and η control the degree of overlapping among the biclusters as well as the degree of coherency.

In order to find out the optimum values of m, ξ, and η, extensive experiments are carried out for yeast microarray data set. Table 10.1 presents the performance of the PBC algorithm, in terms of the average mean squared residue H_{avg}, average volume V_{avg}, average dimensions of the biclusters (I_{avg} and J_{avg}), and degree of overlapping $\mathscr{R}$, for different values of m, ξ and η. Results are presented only for yeast data set. From the results reported in Table 10.1, it is seen that

1. for fixed values of m and η, as the value of ξ increases, the values of H_{avg}, V_{avg}, and $\mathscr{R}$ index decrease;
2. for fixed values of m and ξ, as the value of η increases, the values of these three indices also increase; and
3. for fixed values of ξ and η, as the value of m increases, the values of these three indices also increase.

However, the PBC algorithm provides best result in terms of all these measures for $m = 1.2$, $\xi = 0.60$, and $\eta = 0.25$.

10.5.2 Analysis of Generated Biclusters

For yeast microarray data set, the best ten biclusters generated by the PBC algorithm are analyzed using the Eisen plot [16]. Figure 10.2 (a–j) presents the Eisen plots of best ten biclusters obtained using the PBC algorithm, along with their mean squared

Table 10.2 Functional Enrichment of the PBC Algorithm for BP

Significant GO Term	FDR (%)	False positive	Corrected P-Value
Ribosome biogenesis	0.00	0.00	8.34E−22
Ribonucleoprotein complex biogenesis	0.00	0.00	2.53E−20
Cellular component	0.00	0.00	1.54E−13
Biogenesis translation	0.00	0.00	8.58E−12
Ribosome biogenesis	0.00	0.00	1.45E−11
Ribosome biogenesis	0.00	0.00	6.22E−11
ncRNA processing	0.00	0.00	6.51E−11
Ribonucleoprotein complex biogenesis	0.00	0.00	1.06E−09
Ribosome biogenesis	0.00	0.00	5.80E−08
Oxidation reduction	0.00	0.00	8.40E−08

Table 10.3 Functional Enrichment of the PBC Algorithm for MF

Significant GO Term	FDR (%)	False Positive	Corrected P-Value
Structural constituent of ribosome	0.00	0.00	4.86E−30
	0.00	0.00	2.76E−26
	0.00	0.00	1.55E−21
	0.00	0.00	1.03E−16
	0.00	0.00	1.05E−12
	0.00	0.00	1.37E−12
	0.00	0.00	6.62E−10
	0.00	0.00	5.61E−08
	0.00	0.00	9.87E−08
Oxidoreductase activity	0.00	0.00	9.26E−08

Table 10.4 Functional Enrichment of the PBC Algorithm for CC

Significant GO Term	FDR (%)	False Positive	Corrected P-Value
Cytosolic ribosome	0.00	0.00	6.27E−40
Cytosolic ribosome	0.00	0.00	1.34E−36
Cytosolic ribosome	0.00	0.00	2.64E−27
Ribonucleoprotein complex	0.00	0.00	3.18E−24
Preribosome	0.00	0.00	2.34E−20
Cytosolic ribosome	0.00	0.00	7.76E−17
Cytosolic ribosome	0.00	0.00	1.77E−16
Cytosolic ribosome	0.00	0.00	7.51E−16
Macromolecular complex	0.00	0.00	1.08E−12
Cytosolic ribosome	0.00	0.00	7.55E−12

residue values. All the results reported in Fig. 10.2 (a–j) establish the fact that the PBC algorithm can efficiently identify groups of genes and conditions of coherent values.

To interpret the biological significance of the generated biclusters, the gene ontology (GO) Term Finder is used [5], which is described in Chap. 8. In Tables 10.2,

Table 10.5 Performance of different biclustering algorithms

Algorithms	H_{avg}	V_{avg}	I_{avg}	J_{avg}	$\frac{H_{\text{avg}}}{V_{\text{avg}}}$
PBC	183.67	2742.18	436.02	7.42	0.067
FLOC	187.54	1825.78	195.00	12.80	0.103
CC	204.29	1576.98	166.71	12.09	0.129
SEBI	205.18	209.92	13.61	15.25	0.977

10.3, and 10.4, details of functional enrichment of best ten biclusters obtained by the PBC algorithm for different ontologies are given. From the results reported in Tables 10.2, 10.3, and 10.4, it is seen that the best ten biclusters generated by the PBC algorithm can be assigned to the gene ontology (GO) biological processes (BP), molecular functions (MF), and cellular components (CC) with high reliability in terms of p-value, false discovery rate (FDR), and expected false positives. That is, the PBC algorithm describes accurately the known classification, the one given by the GO, and thus reliable for extracting new biological insights.

10.5.3 Comparative Analysis of Different Methods

In Table 10.5, the performance of the PBC algorithm is compared with that of CC [11], FLOC [51], and SEBI [14], in terms of the average mean squared residue H_{avg}, the average dimensions of the biclusters found (V_{avg}, I_{avg}, and J_{avg}) and the ratio between H_{avg} and V_{avg}.

From the results reported in Table 10.5, it can be seen that the PBC method is capable of finding biclusters with a higher volume than the ones found by the CC, FLOC, and SEBI. This is due to the possibilistic concept adopted by the PBC algorithm. As far as the mean squared residue is concerned, the PBC method is able to find biclusters of relatively lower mean squared residue than that of the CC and SEBI, and comparable mean squared residue with that of the FLOC. Hence, the novel possibilistic biclustering algorithm provides better performance with respect to average mean squared residue as well as average volume of the biclusters found. Also, the ratio between H_{avg} and V_{avg} is minimum in case of the PBC method.

Finally, the performance of three algorithms, namely, PBC, FLOC, and CC is analyzed with respect to gene annotation on yeast microarray data set. The results of best ten biclusters with respect to p-value of different ontology types obtained by three methods are reported in Table 10.6. The results reported in Table 10.6 indicate that the PBC algorithm provides better selectivity compared to that of the FLOC and CC for all of the cases. Only in case of biological processes for tenth bicluster, the CC provides better result than the PBC algorithm.

Table 10.6 Functional enrichment analysis of different methods

Gene Ontology	Different algorithms		
	PBC	Cheng and Church	Yang et al. (FLOC)
Biological processes	8.34E−22	2.49E−20	8.76E−10
	2.53E−20	2.16E−14	2.60E−09
	1.54E−13	1.24E−11	6.24E−09
	8.58E−12	1.34E−10	5.27E−08
	1.45E−11	1.06E−09	1.12E−07
	1.06E−09	3.44E−08	3.25E−07
	5.80E−08	5.50E−08	3.85E−07
	8.40E−08	8.93E−08	4.10E−07
	1.07E−07	9.64E−08	5.33E−07
Molecular functions	4.85E−30	2.09E−15	1.09E−12
	2.76E−26	6.74E−13	1.67E−10
	1.55E−21	1.25E−06	2.83E−09
	1.03E−16	1.79E−06	9.51E−09
	1.05E−12	2.53E−05	1.52E−08
	1.37E−12	5.23E−05	2.43E−07
	6.62E−10	6.00E−05	1.50E−05
	5.61E−08	1.02E−04	5.08E−05
	9.26E−08	1.99E−04	1.02E−04
	9.87E−08	4.00E−04	6.81E−04
Cellular components	6.27E−40	5.00E−21	1.01E−20
	1.34E−36	4.16E−18	1.29E−11
	2.64E−27	1.42E−14	4.43E−11
	3.18E−24	4.90E−12	1.08E−09
	2.34E−20	8.12E−11	1.37E−09
	7.76E−17	2.32E−10	2.14E−08
	1.77E−16	5.13E−10	6.91E−08
	7.51E−16	1.15E−09	1.33E−07
	1.08E−12	1.34E−09	1.84E−07
	7.55E−12	5.77E−09	2.47E−07

10.6 Conclusion and Discussion

In this chapter, the problem of biclustering gene expression data has been addressed. The main difference between conventional clustering and biclustering, and a brief survey on different existing biclustering algorithms have been reported, along with their merits and demerits. A possibilistic biclustering (PBC) algorithm is presented for discovering value-coherent overlapping δ-biclusters. A new quantitative measure is described, along with some existing measures, to evaluate the quality of generated biclusters. Finally, the effectiveness of the PBC algorithm, along with a comparison with existing biclustering algorithms, is demonstrated on yeast gene expression data set.

The concept of possibilistic memberships of an element in different biclusters is found to be successful in efficient selection of highly coherent overlapping biclusters compared to the existing methods. The two-way k-medoids clustering algorithm, based on mutual information, provides a set of small coregulated submatrices as initial biclusters, which facilitates refining biclusters with lesser iteration steps in the final phase. Some of the quantitative indices used for evaluating the quality of selected biclusters may be used in a suitable combination to act as the objective function of an evolutionary algorithm, for generating value-coherent overlapping δ-biclusters.

However, one of the important limitations of the PBC approach, like other biclustering methods, is its execution time. To reduce this weakness, one can integrate the PBC algorithm and theory of rough sets as the incorporation of rough sets into fuzzy and possibilistic clustering reduces the execution time drastically [32]. In the current work, the PBC method is only applied on yeast microarray data set. However, this method can also be applied on other high dimensional microarray data sets and further its merits and limitations can be evaluated.

References

1. Ben-Dor A, Chor B, Karp R, Yakhini Z (2002) Discovering local structure in gene expression data: the order-preserving submatrix problem. In: Proceedings of the 6th international conference on computational biology, pp 49–57
2. Bezdek J (1980) A convergence theorem for the fuzzy ISODATA clustering algorithm. IEEE Trans Pattern Anal Mach Intell 2:1–8
3. Bezdek J, Hathaway RJ, Sabin MJ, Tucker WT (1987) Convergence theory for fuzzy C-means: counterexamples and repairs. IEEE Trans Syst Man Cybern 17:873–877
4. Bezdek JC (1981) Pattern recognition with fuzzy objective function algorithm. Plenum Press, New York
5. Boyle EI, Weng S, Gollub J, Jin H, Botstein D, Cherry JM, Sherlock G (2004) GO:term finder open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. Bioinform 20(18):3710–3715
6. Bryan K, Cunningham P, Bolshakova N (2005) Application of simulated annealing to the biclustering of gene expression data. In: Proceedings of the 18th IEEE symposium on computer-based medical systems, pp 383–388
7. Califano A, Stolovitzky G, Tu Y (2000) Analysis of gene expression microarrays for phenotype classifiation. In: Proceedings of the international conference on computational, molecular biology, pp 75–85
8. Cano C, Adarve L, Lopez J, Blanco A (2007) Possibilistic approach for biclustering microarray data. Comput Biol Med 37:1426–1436
9. Chakraborty A, Maka H (2005) Biclustering of gene expression data using genetic algorithm. In: Proceedings of the IEEE symposium on computational intelligence in bioinformatics and computational biology, pp 1–8
10. Chen G, Sullivan PF, Kosoroka MR (2013) Biclustering with heterogeneous variance. Proc Nat Acad Sci U.S.A 110(30):12253–12258
11. Cheng Y, Church GM (2000) Biclustering of expression data. In: Proceedings of the 8th international conference on intelligent systems for, molecular biology, pp 93–103

12. Cho H, Dhillon I, Guan Y, Sra S (2004) Minimum sum-squared residue coclustering of gene expression data. In: Proceedings of the 4th SIAM international conference on data mining, pp 114–125
13. Das C, Maji P (2013) Possibilistic biclustering algorithm for discovering value-coherent overlapping δ-Biclusters. Int J Mach Learn Cybern. doi:10.1007/s13042-013-0211-3
14. Divina F, Aguilar-Ruiz JS (2006) Biclustering of expression data with evolutionary computation. IEEE Trans Knowl Data Eng 18(5):590–602
15. Domany E (2003) Cluster analysis of gene expression data. J Stat Phys 110(3–6):1117–1139
16. Eisen MB, Spellman PT, Brown PO, Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. Proc Nat Acad Sci U.S.A 95(25):14863–14868
17. Eren K, Deveci M, Kucuktunc O, Catalyurek UV (2012) A comparative analysis of biclustering algorithms for gene expression data. Briefings in bioinformatics. doi:10.1093/bib/bbs032
18. Fei X, Lu S, Pop HF, Liang LR (2007) GFBA: a biclustering algorithm for discovering value-coherent biclusters. Bioinformatics research and applications, pp 1–12
19. Getz G, Levine E, Domany E (2000) Coupled two-way clustering analysis of gene microarray data. Proc Nat Acad Sci U.S.A 97(22):12079–12084
20. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, Lander ES (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286(5439):531–537
21. Hartigan JA (1972) Direct clustering of a data matrix. J Am Stat Assoc 67(337):123–129
22. Hartigan JA, Wong MA (1979) A K-means clustering algorithms. Appl Stat 28:100–108
23. Herrero J, Valencia A, Dopazo J (2001) A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinform 17:126–136
24. James G (1996) Modern engineering mathematics. Addison-Wesley, Reading
25. Jiang D, Tang C, Zhang A (2004) Cluster analysis for gene expression data: a survey. IEEE Trans Knowl Data Eng 16(11):1370–1386
26. Kaufmann L, Rousseeuw PJ (1990) Finding groups in data: an introduction to cluster analysis
27. Krishnapuram R, Keller JM (1993) A possibilistic approach to clustering. IEEE Trans Fuzzy Syst 1(2):98–110
28. Lazzeroni L, Owen A (2000) Plaid models for gene expression data. Technical Report, Standford University
29. Lee M, Shen H, Huang JZ, Marron JS (2010) Biclustering via sparse singular value decomposition. Biometrics 66(4):1087–1095
30. Liu J, Wang W (2003) OP-cluster: clustering by tendency in high dimensional space. In: Proceedings of the 3rd IEEE international conference on data mining, pp 187–194
31. Madeira SC, Oliveira AL (2004) Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Trans Comput Biol Bioinform 1(1):24–45
32. Maji P, Pal SK (2007) Rough set based generalized fuzzy C-means algorithm and quantitative indices. IEEE Trans Syst Man Cybern Part B Cybern 37(6):1529–1540
33. Murali TM, Kasif S (2003) Extracting conserved gene expression motifs from gene expression data. In: Proceedings of the pacific symposium on biocomputing 8:77–88
34. Pawlak Z (1991) Rough sets: theoretical aspects of reasoning about data. Kluwer, Dordrecht
35. Rodriguez-Baena DS, Perez-Pulido AJ, Aguilar-Ruiz JS (2011) A biclustering algorithm for extracting bit-patterns from binary data sets. Bioinform 27(19):2738–2745
36. Segal E, Taskar B, Gasch A, Friedman N, Koller D (2001) Rich probabilistic models for gene expression. Bioinform 17(S1):243–252
37. Sheng Q, Moreau Y, Moor BD (2003) Biclustering microarray data by Gibbs sampling. Bioinform 19(S2):ii196–ii205
38. Sill M, Kaiser S, Benner A, Kopp-Schneider A (2011) Robust biclustering by sparse singular value decomposition incorporating stability selection. Bioinform 27(15):2089–2097
39. Sutheeworapong S, Ota M, Ohta H, Kinoshita K (2012) A novel biclustering approach with iterative optimization to analyze gene expression data. Adv Appl Bioinform Chem 2012(5):23–59

40. Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander ES, Golub TR (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. Proc Nat Acad Sci U.S.A 96(6):2907–2912
41. Tanay A, Sharan R, Shamir R (2002) Discovering statistically significant biclusters in gene expression data. Bioinform 18(S1):136–144
42. Tang C, Zhang L, Zhang A, Ranmanathan M (2001) Interrelated two-way clustering: an unsupervised approach for gene expression data analysis. In: Proceedings of the 2nd IEEE international symposium on bioinformatics and bioengineering, pp 41–48
43. Tibshirani R, Hastie T, Eisen M, Ross D, Bostein D, Brown P (1999) Clustering methods for the analysis of DNA microarray data. Technical Report, Standford University
44. Tjhi WC, Chen L (2006) A partitioning based algorithm to fuzzy co-cluster documents and words. Pattern Recogn Lett 27:151–159
45. Tjhi WC, Chen L (2007) Possibilistic fuzzy co-clustering of large document collections. Pattern Recogn 40:3452–3466
46. Tjhi WC, Chen L (2008) A heuristic based fuzzy co-clustering algorithm for categorization of high dimensional data. Fuzzy Sets Syst 159:371–389
47. Tjhi WC, Chen L (2008) Dual fuzzy-possibilistic co-clustering for categorization of documents. IEEE Trans Fuzzy Syst 17(3):532–543
48. Wang R, Miao D, Li G, Zhang H (2007) Rough overlapping biclustering of gene expression data. In: Proceedings of the 7th IEEE international conference on bioinformatics and bioengineering, pp 828–834
49. Wu CJ, Fu Y, Murali TM, Kasif S (2004) Gene expression module discovery using Gibbs sampling. Genome Inf 15(1):239–248
50. Yan H (2004) Convergence condition and efficient implementation of the fuzzy curve-tracing (FCT) algorithm. IEEE Trans Syst Man Cybern Part B Cybern 34(1):210–221
51. Yang J, Wang W, Wang H, Yu PS (2003) Enhanced biclustering on expression data. In: Proceedings of the 3rd IEEE international conference on bioinformatics and bioengineering, pp 321–327
52. Zadeh LA (1965) Fuzzy sets. Inf Control 8:338–353

Chapter 11
Fuzzy Measures and Weighted Co-Occurrence Matrix for Segmentation of Brain MR Images

11.1 Introduction

In medical imaging technology, a number of complementary diagnostic tools such as X-ray computer tomography, position emission tomography, and magnetic resonance imaging (MRI) are available. The MRI is an important diagnostic imaging technique for the early detection of abnormal changes in tissues and organs. Its unique advantage over other modalities is that it can provide multispectral images of tissues with a variety of contrasts based on three MR parameters, namely, ρ, T1, and T2. Therefore, majority of research in medical image analysis concerns the MR images [36].

Conventionally, these images are interpreted visually and qualitatively by radiologists. Advanced research requires quantitative information such as the size of the brain ventricles after a traumatic brain injury or the relative volume of ventricles to brain. Fully automatic methods sometimes fail, producing incorrect results and requiring the intervention of a human operator. This is often true due to restrictions imposed by image acquisition, pathology, and biological variation. Hence, it is important to have a faithful method to measure various structures in the brain. One of such methods is the segmentation of images to isolate objects and regions of interest.

Image segmentation is an indispensable process in the visualization of human tissues, particularly during clinical analysis of medical images. In the analysis of medical images for computer-aided diagnosis and therapy, segmentation is often required as a preliminary stage. The success of an image analysis system depends on the quality of segmentation [32, 33, 36]. Medical image segmentation is a complex and challenging task due to the intrinsic nature of the images. The brain has a particularly complicated structure and its precise segmentation is very important for detecting tumors, edema, and necrotic tissues, in order to prescribe appropriate therapy [32, 33, 36].

Segmentation is a process of partitioning an image space into some nonoverlapping meaningful homogeneous regions. If the domain of the image is given by Ω, then the segmentation problem is to determine the sets $S_k \subset \Omega$, whose union is the entire domain Ω. The sets that make up a segmentation must satisfy

P. Maji and S. Paul, *Scalable Pattern Recognition Algorithms*,
DOI: 10.1007/978-3-319-05630-2_11,

$$\Omega = \bigcup_{k=1}^{k} S_k, \tag{11.1}$$

where $S_k \cap S_j = \emptyset$ for $k \neq j$, and each S_k is connected. Hence, a segmentation method is supposed to find those sets that correspond to distinct anatomical structures or regions of interest in the image.

Many image processing techniques have been proposed for the MR image segmentation [2, 7], most notably thresholding [10, 14, 34], region-growing [18], edge detection [35], pixel classification [25, 31], and clustering [1, 11, 40]. Some algorithms using the neural network approach have been investigated in the MR image segmentation problems [5, 6]. The segmentation of the MR images using fuzzy c-means has been reported in [1, 4, 6, 12, 28, 45]. Image segmentation using rough sets has also been done in [8, 16, 17, 19, 30, 42–44]. Recently, a review is reported in [9] on the application of rough sets and near sets in medical imaging.

Thresholding is one of the old, simple, and popular techniques for image segmentation. It can be done based on global (for example, gray level histogram of the entire image) or local information (for example, co-occurrence matrix) extracted from the image. A series of algorithms for image segmentation based on histogram thresholding can be found in the literature [2, 7, 10, 14, 20, 25, 34, 37]. Entropy-based thresholding algorithms have been proposed in [21–23, 29]. One of the main problems in medical image segmentation is uncertainty. Some of its sources include imprecision in computations and vagueness in class definitions. In this background, the possibility concept introduced by the fuzzy set theory has gained popularity in modeling and propagating uncertainty in medical imaging applications [15, 16]. Also, since the fuzzy set theory is a powerful tool to deal with linguistic concepts such as similarity, several segmentation algorithms based on fuzzy set theory are reported in the literature [3, 13, 26, 29, 38].

In general, all histogram thresholding techniques based on fuzzy set theory work very well when the image gray level histogram is bimodal or multimodal. On the other hand, a great deal of medical images are usually unimodal, where the conventional histogram thresholding techniques perform poorly or even fail. In this class of histograms, unlike the bimodal case, there is no clear separation between object and background pixel occurrences. Hence, to find a reliable threshold, some adequate criteria for splitting the image histogram should be used. In [38], an approach to threshold the histogram according to the similarity between gray levels has been proposed.

This chapter presents a new algorithm, termed as the FMWCM [13, 14], to threshold the image histogram. It is based on a fuzzy measure and the concept of weighted co-occurrence matrix. The second order fuzzy measures such as fuzzy correlation, fuzzy entropy, and index of fuzziness, are used for assessing such a concept. The local information of the given image is extracted through a modified co-occurrence matrix. The FMWCM technique consists of two linguistic variables {bright, dark} modeled by two fuzzy subsets and a fuzzy region on the gray level histogram. Each of the gray levels of the fuzzy region is assigned to both defined subsets one by one

and the second order fuzzy measure using weighted co-occurrence matrix is calculated. The ambiguity of each gray level is determined from the fuzzy measures of two fuzzy subsets. Finally, the strength of ambiguity for each gray level is computed. The multiple thresholds of the image histogram are determined according to the strength of ambiguity of the gray levels using a nearest mean classifier. Experimental results reported in this chapter confirm that the FMWCM method is robust in segmenting brain MR images compared to existing popular thresholding techniques.

The rest of this chapter is as follows: In Sect. 11.2, some basic definitions about fuzzy sets and second order fuzzy measures along with co-occurrence matrix are reported. The FMWCM algorithm for histogram thresholding is presented in Sect. 11.3. Experimental results and a comparison with other thresholding methods are presented in Sect. 11.4. Concluding remarks are given in Sect. 11.5.

11.2 Fuzzy Measures and Co-Occurrence Matrix

This section presents the basic notions in the theory of fuzzy sets and the concept of co-occurrence matrix, along with different second order fuzzy measures and fuzzy membership function.

11.2.1 Fuzzy Set

A fuzzy subset A of the universe X is defined as a collection of ordered pairs

$$A = \{(\mu_A(x), x), \forall x \in X\} \tag{11.2}$$

where $\mu_A(x)$ denotes the degree of belonging of the element x to the fuzzy set A and $0 \leq \mu_A(x) \leq 1$. The support of fuzzy set A is the crisp set that contains all the elements of X that have a nonzero membership value in A [46].

Let $X = [x_{mn}]$ be an image of size $M \times N$ and L gray levels, where x_{mn} is the gray value at location (m, n) in X, $x_{mn} \in G_L$, $G_L = \{0, 1, 2, ..., L-1\}$ is the set of the gray levels, $m = 0, 1, 2, \cdots, M-1, n = 0, 1, 2, \cdots, N-1$, and $\mu_X(x_{mn})$ be the value of the membership function in the unit interval $[0, 1]$, which represents the degree of possessing some brightness property $\mu_X(x_{mn})$ by the pixel intensity x_{mn}. By mapping an image X from x_{mn} into $\mu_X(x_{mn})$, the image set X can be written as

$$X = \{\mu_X(x_{mn}), x_{mn}\}. \tag{11.3}$$

Then, X can be viewed as a characteristic function and μ_X is a weighting coefficient that reflects the ambiguity in X. A function mapping all the elements in a crisp set into real numbers in $[0, 1]$ is called a membership function. The larger value of the membership function represents the higher degree of the membership. It means

how closely an element resembles an ideal element. Membership functions can represent the uncertainty using some particular functions. These functions transform the linguistic variables into numerical calculations by setting some parameters. The fuzzy decisions can then be made. The standard S-function, that is, $S(x_{mn}; a, b, c)$, of Zadeh is as follows [46]:

$$\mu_X(x_{mn}) = \begin{cases} 0 & x_{mn} \leq a \\ 2\left[\frac{x_{mn}-a}{c-a}\right]^2 & a \leq x_{mn} \leq b \\ 1 - 2\left[\frac{x_{mn}-c}{c-a}\right]^2 & b \leq x_{mn} \leq c \\ 1 & x_{mn} \geq c \end{cases} \tag{11.4}$$

where $b = \frac{(a+c)}{2}$ is the crossover point for which the membership value is 0.5. The shape of S-function is manipulated by the parameters a and c.

11.2.2 Co-Occurrence Matrix

The co-occurrence matrix or the transition matrix of the image X is an $L \times L$ dimensional matrix that gives an idea about the transition of intensity between adjacent pixels. In other words, the (i, j)th entry of the matrix gives the number of times the gray level j follows the gray level i, that is, the gray level j is an adjacent neighbor of the gray level i, in a specific fashion. Let a be the (m, n)th pixel in X and b denotes one of the eight neighboring pixels of a, that is,

$$b \in a_8 = \{(m, n-1), (m, n+1), (m+1, n), (m-1, n), (m-1, n-1), (m-1, n+1), (m+1, n-1), (m+1, n+1)\}$$

$$\text{then } t_{ij} = \sum_{\substack{a \in X \\ b \in a_8}} \delta; \tag{11.5}$$

$$\text{where } \delta = \begin{cases} 1 \text{ if gray level value of } a \text{ is } i \text{ and that of } b \text{ is } j \\ 0 \text{ otherwise.} \end{cases} \tag{11.6}$$

Obviously, t_{ij} gives the number of times the gray level j follows gray level i in any one of the eight directions. The matrix $T = [t_{ij}]_{L \times L}$ is, therefore, the co-occurrence matrix of the image X.

11.2.3 Second Order Fuzzy Correlation

The correlation between two local properties μ_1 and μ_2 (for example, edginess, blurredness, and texture) can be expressed in the following ways [27]:

$$C(\mu_1, \mu_2) = 1 - \frac{4\sum_{i=1}^{L}\sum_{j=1}^{L}[\mu_1(i,j) - \mu_2(i,j)]^2 t_{ij}}{Y_1 + Y_2} \tag{11.7}$$

where t_{ij} is the frequency of occurrence of the gray level i followed by j, that is, $T = [t_{ij}]_{L\times L}$ is the co-occurrence matrix defined earlier, and

$$Y_k = \sum_{i=1}^{L}\sum_{j=1}^{L}[2\mu_k(i,j) - 1]^2 t_{ij}; \quad k = 1, 2. \tag{11.8}$$

To calculate the correlation between a gray-tone image and its two-tone version, μ_2 is considered as the nearest two-tone version of μ_1, that is,

$$\mu_2(x) = \begin{cases} 0 \text{ if } \mu_1(x) \le 0.5 \\ 1 \text{ otherwise.} \end{cases} \tag{11.9}$$

11.2.4 Second Order Fuzzy Entropy

Out of the n pixels of the image X, consider a combination of r elements. Let S_i^r be the ith such combination and $\mu(S_i^r)$ denotes the degree to which the combination S_i^r, as a whole, possesses the property μ. There are $\binom{n}{r}$ such combinations. The entropy of order r of the image X is defined as [24]

$$H^{(r)} = -\frac{1}{N}\sum_{i=1}^{N}\left[\mu(S_i^r)\ln\{\mu(S_i^r)\} + \{1 - \mu(S_i^r)\}\ln\{1 - \mu(S_i^r)\}\right] \tag{11.10}$$

with logarithmic gain function and $N = \binom{n}{r}$. It provides a measure of the average amount of difficulty or ambiguity in making a decision on any subset of r elements as regards to its possession of an imprecise property. Normally, these r pixels are chosen as adjacent pixels. For the present investigation, the value of r is chosen as 2.

11.2.5 Second Order Index of Fuzziness

The quadratic index of fuzziness of an image X of size $M \times N$ reflects the average amount of ambiguity or fuzziness present in it by measuring the distance (quadratic) between its fuzzy property plane μ_1 and the nearest two-tone version μ_2. In other words, the distance between the gray-tone image and its nearest two-tone version [29]. If we consider spatial information in the membership function, then the index of fuzziness takes the form

$$I(\mu_1, \mu_2) = \frac{2\left\{\sum_{i=1}^{L}\sum_{j=1}^{L}[\mu_1(i,j) - \mu_2(i,j)]^2 t_{ij}\right\}^{\frac{1}{2}}}{\sqrt{MN}} \tag{11.11}$$

where t_{ij} is the frequency of occurrence of the gray level i followed by j.

For computing the second order fuzzy measures such as correlation, entropy, and index of fuzziness of an image, represented by a fuzzy set, one needs to choose two pixels at a time and to assign a composite membership value to them. Normally these two pixels are chosen as adjacent pixels.

11.2.6 2D S-Type Membership Function

This section presents a two dimensional S-type membership function that represents fuzzy bright image plane assuming higher gray value corresponds to object region. The 2D S-type membership function reported in [28] assigns a composite membership value to a pair of adjacent pixels as follows: For a particular threshold b,

1. (b, b) is the most ambiguous point, that is, the boundary between object and background. Therefore, its membership value for the fuzzy bright image plane is 0.5.
2. If one object pixel is followed by another object pixel, then its degree of belonging to object region is greater than 0.5. The membership value increases with increase in pixel intensity.
3. If one object pixel is followed by one background pixel or vice versa, the membership value is less than or equal to 0.5, depending on the deviation from the boundary point (b, b).
4. If one background pixel is followed by another background pixel, then its degree of belonging to object region is less than 0.5. The membership value decreases with decrease of pixel intensity.

Instead of using fixed bandwidth (Δb), the parameters of S-type membership function are taken as follows [38]:

$$b = \frac{\sum_{i=p}^{q} x_i \cdot h(x_i)}{\sum_{i=p}^{q} h(x_i)}; \tag{11.12}$$

$$\Delta b = \max\{|b - (x_i)_{\min}|, |b - (x_i)_{\max}|\}; \tag{11.13}$$

$$c = b + \Delta b; \tag{11.14}$$

$$a = b - \Delta b; \tag{11.15}$$

where $h(x_i)$ denotes the image histogram, and x_p and x_q are the limits of the subset being considered. The quantities $(x_i)_{\min}$ and $(x_i)_{\max}$ represent the minimum and maximum gray levels in the current set for which $h((x_i)_{\min}) \neq 0$ and $h((x_i)_{\max}) \neq 0$, respectively. Basically, the crossover point b is the mean gray level value of the interval $[x_p, x_q]$. With the function parameters computed in this way, the S-type membership function adjusts its shape as a function of the set elements.

11.3 Thresholding Algorithm

The FMWCM method, proposed by Maji et al. [13, 14] for segmentation of brain MR images, consists of three phases, namely,

1. modification of co-occurrence matrix;
2. measure of ambiguity for each gray level x_i; and
3. measure of strength of ambiguity.

Each of the three phases is elaborated next one by one.

11.3.1 Modification of Co-Occurrence Matrix

In general, for a given image consisting of object on a background, the object and background each have a unimodal gray level population. The gray levels of adjacent points interior to the object, or to the background, are highly correlated, while across the edges at which object and background meet, adjacent points differ significantly in gray level. If an image satisfies these conditions, its gray level histogram will be primarily a mixture of two unimodal histograms corresponding to the object

and background populations, respectively. If the means of these populations are sufficiently far apart, their standard deviations are sufficiently small, and they are comparable in size, the image histogram will be bimodal.

In medical imaging, the histogram of the given image is in general unimodal. One side of the peak may display a shoulder or slope change, or one side may be less steep than the other, reflecting the presence of two peaks that are close together or that differ greatly in height. The histogram may also contain a third, usually smaller, population corresponding to points on the object-background border. These points have gray levels intermediate between those of the object and background; their presence raises the level of the valley floor between the two peaks, or if the peaks are already close together, makes it harder to detect the fact that they are not a single peak.

As the histogram peaks are close together and very unequal in size, it may be difficult to detect the valley between them. This chapter presents a method of producing a transformed co-occurrence matrix in which the valley is deeper and is thus easier to detect. In determining how each point of the image should contribute to the transformed co-occurrence matrix, the FMWCM method takes into account the rate of change of gray level at the point, as well as the point's gray level (edge value); that is, the maximum of differences of average gray levels in pairs of horizontally and vertically adjacent 2-by-2 neighborhoods [41]. If Δ is the edge value at a given point, then

$$\begin{aligned}\Delta = \frac{1}{4}\max\{&|x_{m-1,n} + x_{m-1,n+1} + x_{m,n} + x_{m,n+1} - x_{m+1,n} - x_{m+1,n+1} \\ &- x_{m+2,n} - x_{m+2,n+1}|, |x_{m,n-1} + x_{m,n} + x_{m+1,n-1} \\ &+ x_{m+1,n} - x_{m,n+1} - x_{m,n+2} - x_{m+1,n+1} - x_{m+1,n+2}|\}. \end{aligned} \tag{11.16}$$

According to the image model, points interior to the object and background should generally have low edge values, since they are highly correlated with their neighbors, while those on the object-background border should have high edge values. Hence, if we produce a co-occurrence matrix of the gray levels of points having low edge values only, the peaks should remain essentially same, since they correspond to interior points, but valley should become deeper, since the intermediate gray level points on the object-background border have been eliminated.

More generally, we can compute a weighted co-occurrence matrix in which points having low edge values are counted heavily, while points having high values are counted less heavily. If $|\Delta|$ is the edge value at a given point, then (11.5) becomes

$$t_{ij} = \sum_{\substack{a \in X \\ b \in a_8}} \frac{\delta}{(1 + |\Delta|^2)}. \tag{11.17}$$

This gives full weight, that is 1, to points having zero edge value and negligible weight to high edge value points.

11.3.2 Measure of Ambiguity

The aim of the FMWCM method is to threshold the gray level histogram by splitting the image histogram into multiple crisp subsets using a second order fuzzy measure such as fuzzy correlation, fuzzy entropy, or index of fuzziness. First, let us define two linguistic variables {dark, bright} modeled by two fuzzy subsets of X, denoted by A and B, respectively. The fuzzy subsets A and B are associated with the histogram intervals $[x_{\min}, x_p]$ and $[x_q, x_{\max}]$, respectively, where x_p and x_q are the final and initial gray level limits for these subsets, and $x_{\min}$ and $x_{\max}$ are the lowest and highest gray levels of the image, respectively. Then, the ratio of the cardinalities of two fuzzy subsets A and B is given by

$$\beta = \frac{n_A}{n_B} = \frac{|\{x_{\min}, x_{\min+1}, \cdots, x_{p-1}, x_p\}|}{|\{x_q, x_{q+1}, \cdots, x_{\max-1}, x_{\max}\}|}. \tag{11.18}$$

Next, we calculate $F_A(x_{\min} : x_p)$ and $F_B(x_q : x_{\max})$, where $F_A(x_{\min} : x_p)$ is the second order fuzzy measure of fuzzy subset A and its two-tone version; and $F_B(x_q : x_{\max})$ is the second order fuzzy measure of fuzzy subset B and its two-tone version using the weighted co-occurrence matrix. Since the key of the FMWCM method is the comparison of fuzzy measures, we have to normalize those measures. This is done by computing a normalizing factor α according to the following relation:

$$\alpha = \frac{F_A(x_{\min} : x_p)}{F_B(x_q : x_{\max})}. \tag{11.19}$$

To obtain the segmented version of the gray level histogram, we add to each of the subsets A and B a gray level x_i picked up from the fuzzy region and form two fuzzy subsets $\acute{A}$ and $\acute{B}$ which are associated with the histogram intervals $[x_{\min}, x_i]$ and $[x_i, x_{\max}]$, where $x_p < x_i < x_q$. Then, we calculate $F_{\acute{A}}(x_{\min} : x_i)$ and $F_{\acute{B}}(x_i : x_{\max})$. The ambiguity of the gray value of x_i is calculated as follows:

$$\mathscr{A}(x_i) = 1 - \frac{|F_{\acute{A}}(x_{\min} : x_i) - \alpha \cdot F_{\acute{B}}(x_i : x_{\max})|}{(1 + \alpha)}. \tag{11.20}$$

Finally, applying this procedure for all gray levels of the fuzzy region, we calculate the ambiguity of each gray level. The process is started with $x_i = x_p + 1$, and x_i is incremented one by one until $x_i > x_q$. The ratio of the cardinalities of two modified fuzzy subsets $\acute{A}$ and $\acute{B}$ at each iteration is being modified accordingly

$$\acute{\beta} = \frac{n_{\acute{A}}}{n_{\acute{B}}} = \frac{|\{x_{\min}, x_{\min+1}, \cdots, x_{i-1}, x_i\}|}{|\{x_i, x_{i+1}, \cdots, x_{\max-1}, x_{\max}\}|} > \beta. \tag{11.21}$$

Unlike [38], in the FMWCM method as x_i is incremented one by one, the value of $\acute{\beta}$ also increases. Figure 11.1 represents the ambiguity $\mathscr{A}(x_i)$ of the gray level x_i

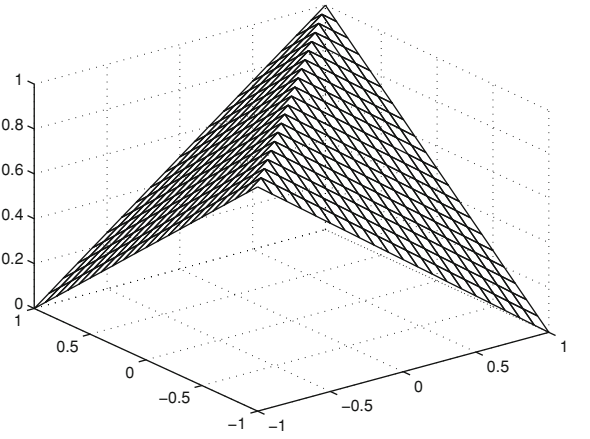

Fig. 11.1 Ambiguity $\mathscr{A}$ of the gray level x_i as a function of fuzzy measures of two fuzzy subsets for $\alpha = 1.0$

as a function of fuzzy measures of two modified fuzzy subsets $\acute{A}$ and $\acute{B}$ for $\alpha = 1.0$. In other words, we calculate the ambiguity by observing how the introduction of a gray level x_i of the fuzzy region affects the similarity measure among gray levels in each of the modified fuzzy subsets $\acute{A}$ and $\acute{B}$. The ambiguity $\mathscr{A}$ is maximum for the gray level x_i in which the fuzzy measures of two modified fuzzy subsets are equal. The threshold level (T) for segmentation corresponds to gray value with maximum ambiguity $\mathscr{A}$. That is,

$$\mathscr{A}(T) = \max \ \arg\{\mathscr{A}(x_i)\}; \ \ \forall \ x_p < x_i < x_q. \tag{11.22}$$

To find out multiple thresholds corresponding to multiple segments, the concept of strength of ambiguity is reported next.

11.3.3 Strength of Ambiguity

In this subsection, the strength of ambiguity ($\mathscr{S}$) of each gray level x_i is calculated as follows. Let, the difference of the gray levels between the current gray level x_i and the gray level x_j, i.e., the closest gray level on the left-hand side whose ambiguity value is larger than or equal to the current ambiguity value is given by

$$\Delta L(x_i) = \begin{cases} x_i - x_j & \text{if } \mathscr{A}(x_j) \geq \mathscr{A}(x_i) \\ 0 & \text{otherwise.} \end{cases} \tag{11.23}$$

Similarly, the difference of the gray levels between the current gray level x_i and the gray level x_k, i.e., the closest gray level on the right-hand side whose ambiguity value is larger than or equal to current ambiguity value is given by

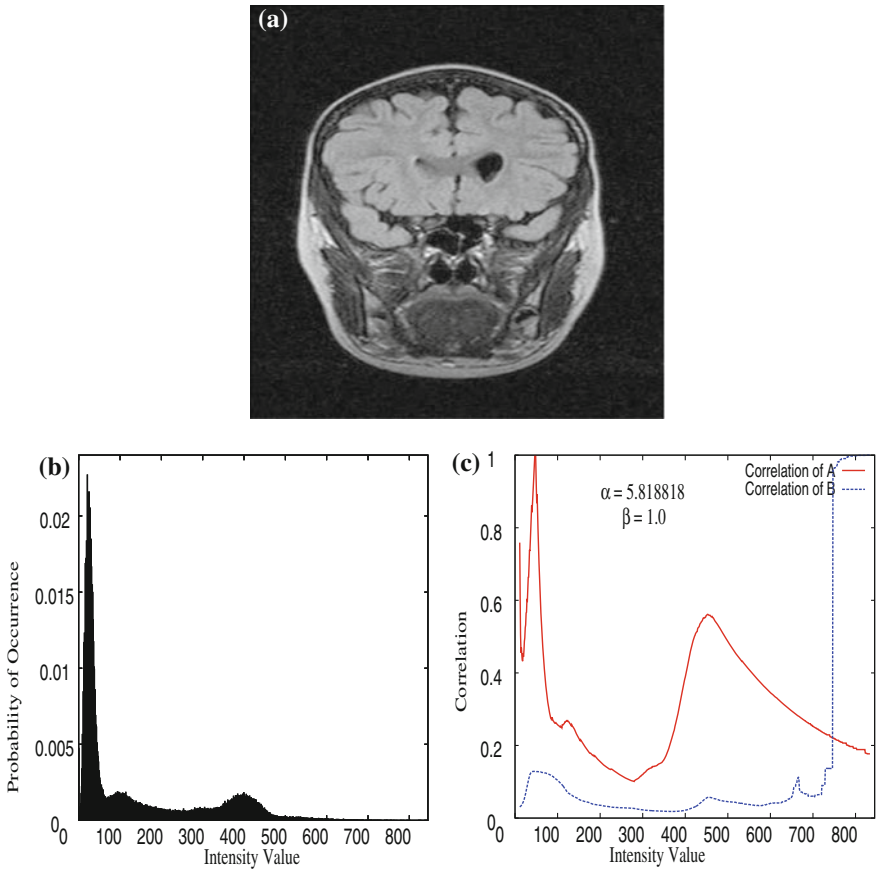

Fig. 11.2 **a** Image I-733; **b** Histogram of given image; and **c** Correlations of two modified fuzzy subsets

$$\Delta R(x_i) = \begin{cases} x_k - x_i & \text{if } \mathscr{A}(x_k) \geq \mathscr{A}(x_i) \\ 0 & \text{otherwise.} \end{cases} \tag{11.24}$$

The strength of ambiguity of the gray level x_i is given by

$$\mathscr{S}(x_i) = \mathscr{D}(x_i) \times \Delta\mathscr{A}(x_i) \tag{11.25}$$

where $\mathscr{D}(x_i)$ is the absolute distance of the gray level x_i and $\Delta\mathscr{A}(x_i)$ is the difference of ambiguities of gray levels x_i and x_m, which is given by

$$\Delta\mathscr{A}(x_i) = \mathscr{A}(x_i) - \mathscr{A}(x_m) \tag{11.26}$$

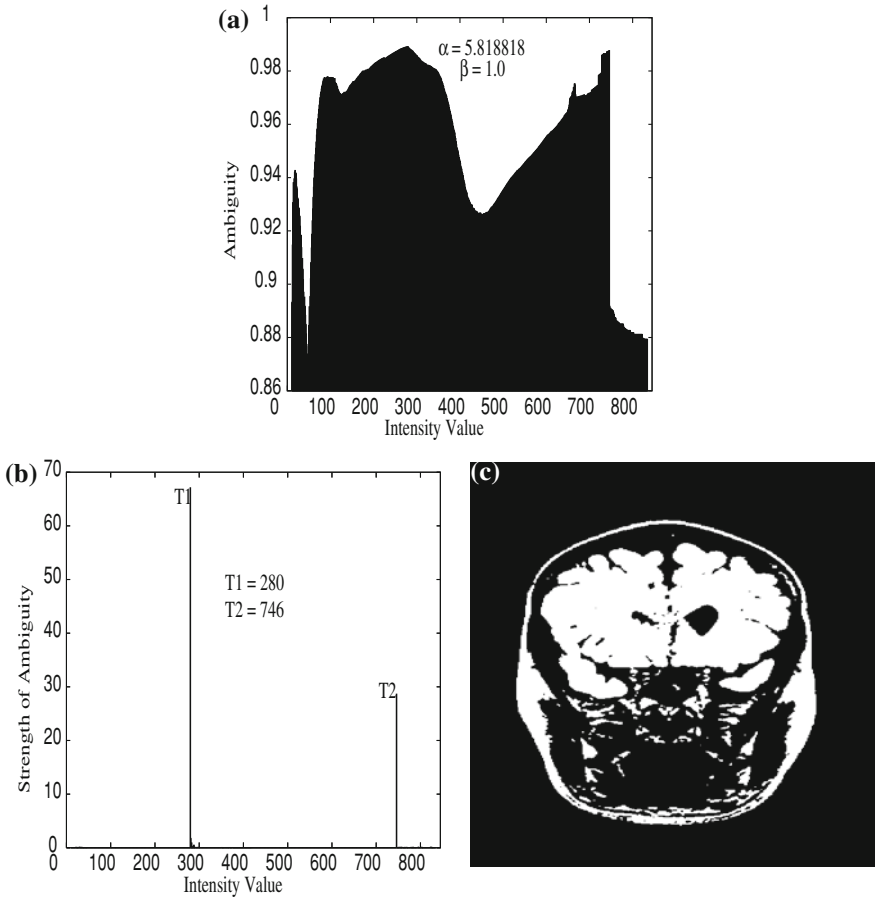

Fig. 11.3 **a** Measure of ambiguity; **b** Strength of ambiguity; and **c** Segmented image obtained using the FMWCM

1. If $\Delta L(x_i) = 0$ and $\Delta R(x_i) = 0$, it means that the current gray level x_i has the highest ambiguity value; then

$$\mathscr{D}(x_i) = \max(x_i - x_{\min}, x_{\max} - x_i) \tag{11.27}$$

   and x_m is the gray level with smallest ambiguity value between $x_{\min}$ and $x_{\max}$.
2. If $\Delta L(x_i) \neq 0$ and $\Delta R(x_i) = 0$, then

$$\mathscr{D}(x_i) = \Delta L(x_i) \tag{11.28}$$

   and x_m is the gray level with smallest ambiguity value between x_i and x_j.
3. If $\Delta R(x_i) \neq 0$ and $\Delta L(x_i) = 0$, then

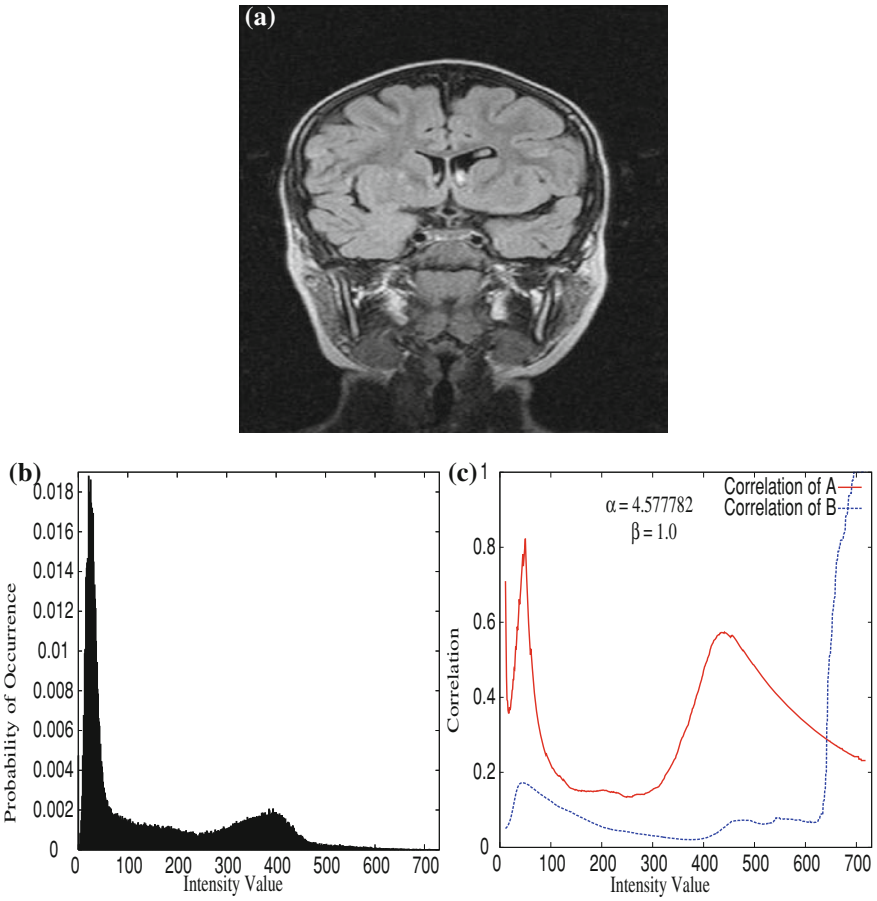

Fig. 11.4 **a** Image I-734; **b** Histogram of given image; and **c** Correlations of two modified fuzzy subsets

$$\mathscr{D}(x_i) = \Delta R(x_i) \tag{11.29}$$

and x_m is the gray level with smallest ambiguity value between x_i and x_k.

4. If $\Delta L(x_i) \neq 0$ and $\Delta R(x_i) \neq 0$, then

$$\mathscr{D}(x_i) = \min(\Delta L(x_i), \Delta R(x_i)), \tag{11.30}$$

and x_m is the gray level with smallest ambiguity value between x_i and x_p, where x_p is the adjacent peak location of the current gray level x_i, where

$$x_p = \begin{cases} x_k \text{ if } \mathscr{A}(x_j) \geq \mathscr{A}(x_k) \\ x_j \text{ otherwise.} \end{cases} \tag{11.31}$$

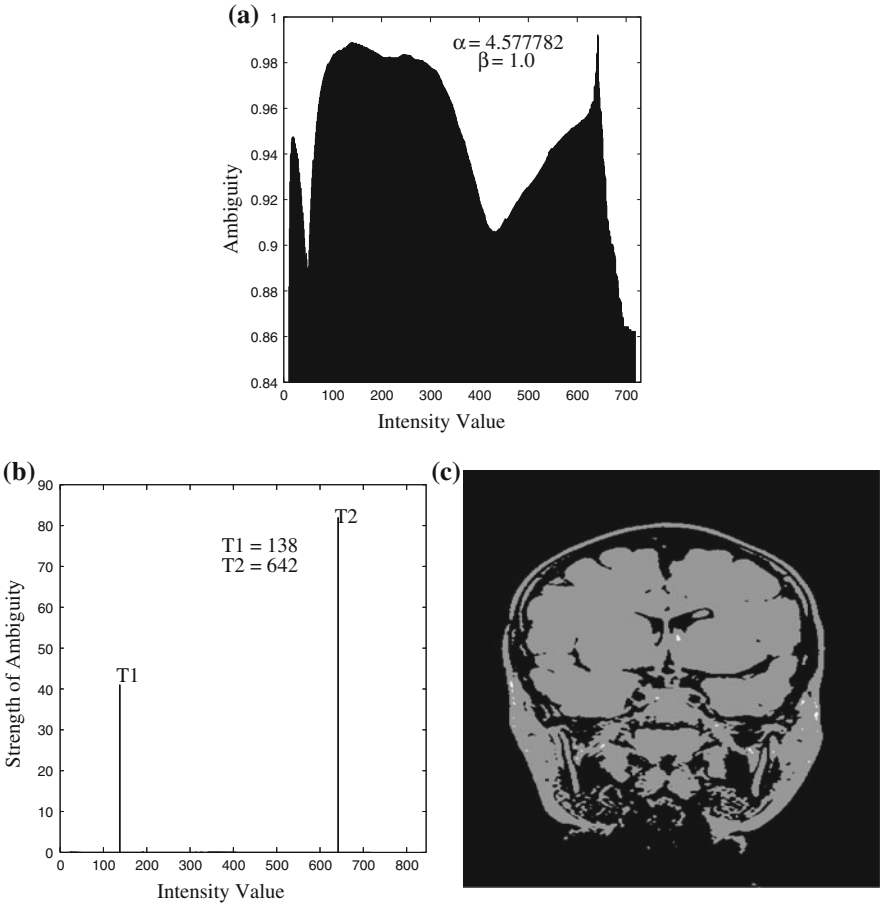

Fig. 11.5 **a** Measure of ambiguity; **b** Strength of ambiguity; and **c** Segmented image obtained using the FMWCM

The thresholds are determined according to the strengths of ambiguity of the gray levels using a nearest mean classifier [39]. If the strength of ambiguity of gray level x_t is the strongest, then x_t is declared to be the first threshold. In order to find other thresholds, $\mathscr{S}(x_t)$ and those strengths of ambiguity which are less than $\mathscr{S}(x_t)/10$ are removed. Then, the mean (M) of strengths of ambiguity is calculated. Finally, the minimum mean distance is calculated as follows:

$$D(x_s) = \min |\mathscr{S}(x_i) - M|; \quad x_p < x_i < x_q \tag{11.32}$$

where x_s is the location that has the minimum distance with M. The strengths that are larger than or equal to $\mathscr{S}(x_s)$ are also declared to be thresholds.

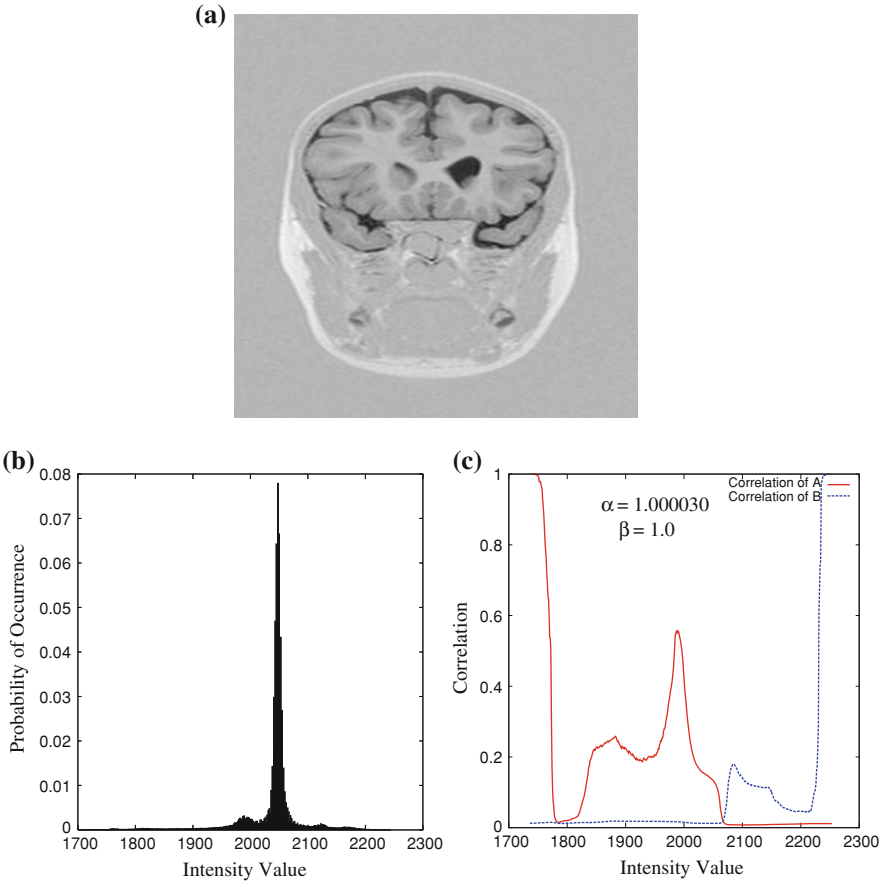

Fig. 11.6 **a** Image I-761; **b** Histogram of given image; and **c** Correlations of two modified fuzzy subsets

11.4 Experimental Results

In this section, the results of different thresholding methods for segmentation of brain MR images are presented. Above 100 MR images with different size and 16 bit gray levels are tested with different methods. All the methods are implemented in C language and run in LINUX environment having machine configuration Pentium IV, 3.2 GHz, 1 MB cache, and 1 GB RAM. All the medical images are brain MR images, which are collected from Advanced Medicare and Research Institute, Kolkata, India.

From (11.18), it is seen that the choice of n_A, n_B, and β is critical. If n_A and n_B increase, the computational time decreases, resulting in nonacceptable segmentation. However, extensive experimentation shows that the typical value of β is 1.0 and $n_A = n_B = 10$ for obtaining acceptable segmentation.

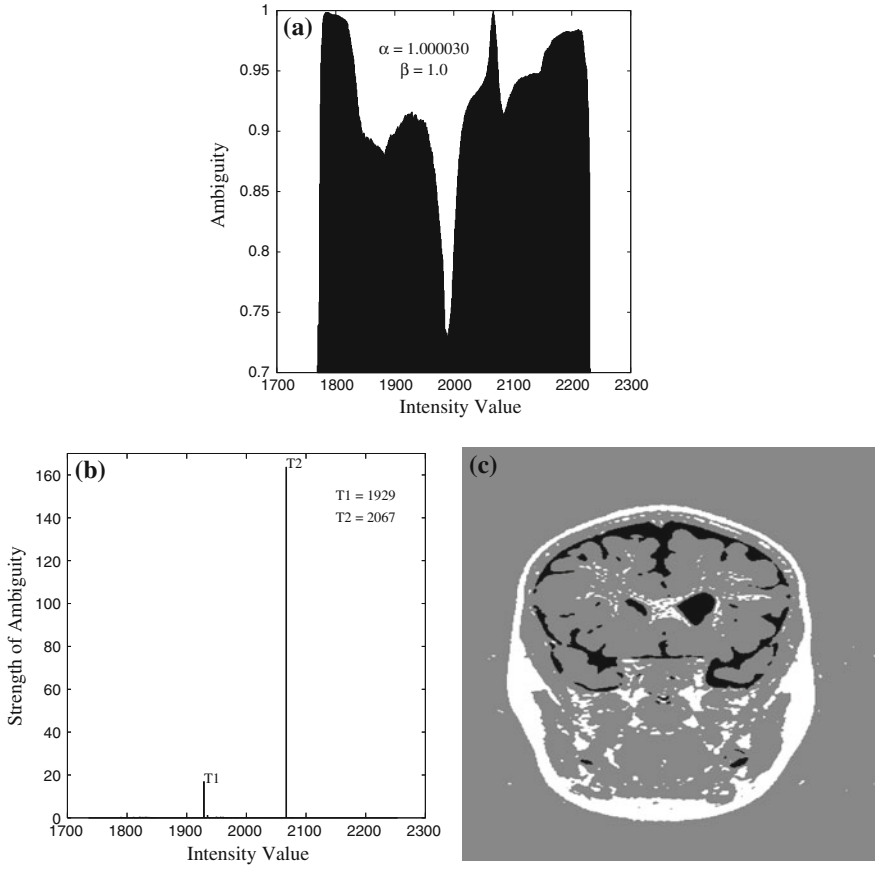

Fig. 11.7 **a** Measure of ambiguity; **b** Strength of ambiguity; and **c** Segmented image obtained using the FMWCM

The FMWCM method is explained using Figs. 11.2–11.7. Figures 11.2, 11.4, and 11.6 show three brain MR images (I-733, I-734, and I-761) and their gray value histograms, along with the second order fuzzy correlations $C_{\acute{A}}(x_{\min} : x_i)$ and $C_{\acute{B}}(x_i : x_{\max})$ of two modified fuzzy subsets $\acute{A}$ and $\acute{B}$ with respect to the gray level x_i of the fuzzy region. The values of α and β are also given in these figures. In Figs. 11.3, 11.5, and 11.7a and b depict the ambiguity and strength of ambiguity of each gray level x_i. The thresholds are determined according to the strength of ambiguity. Finally, Figs. 11.3c, 11.5c, and 11.7c show the segmented images obtained using the FMWCM method. The multiple thresholds obtained using three fuzzy measures such as fuzzy correlation (2-DFC), fuzzy entropy (2-DEntropy), and index of fuzziness (2-DIOF) for these three images (I-733, I-734, and I-761) are reported in Table 11.1. The results reported in Table 11.1 establish the fact that the FMWCM method is independent of the fuzzy measures used such as fuzzy correlation, fuzzy

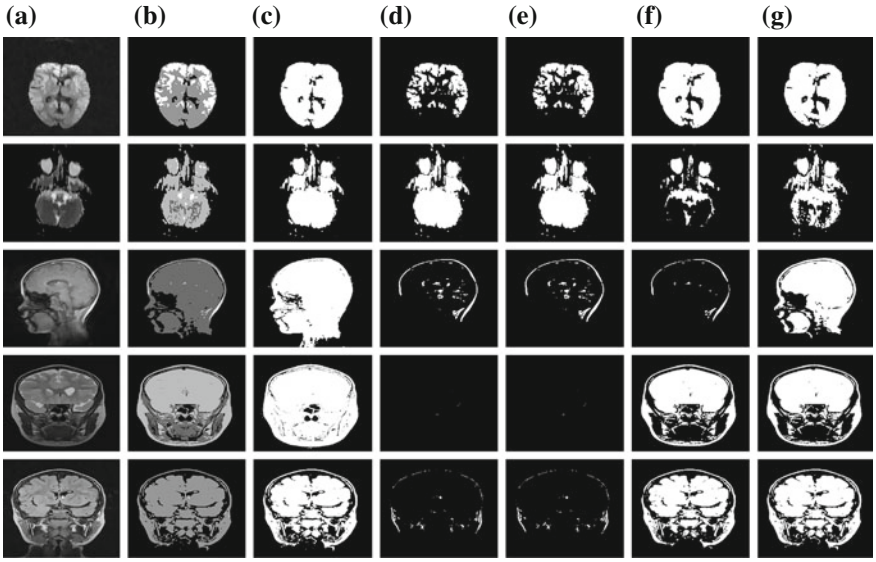

Fig. 11.8 Brain MR images (*top* to *bottom*: I-629, I-647, I-677, I-704, I-734) along with the segmented images **a** Image ; **b** FMWCM ; **c** IOAC ; **d** 1-DFC ;**e** 2-DFC ; **f** Entropy and **g** Otsu

Table 11.1 Results on I-733, I-734, and I-761

Image	Size	Gray	Gray value		Thresholds		
Index	$(M \times N)$	Level	Maximum	Minimum	2-DFC	2-DEntropy	2-DIOF
I-733	512×360	843	842	0	280, 746	278, 743	280, 746
I-734	512×360	730	729	0	138, 642	137, 640	134, 640
I-761	512×360	540	2244	1705	1929, 2067	1930, 2064	1930, 2067

entropy, and index of fuzziness. In all these cases, the number of thresholds are same and the threshold values are very close to each other.

The comparative segmentation results of different thresholding techniques are presented next. Table 11.2 represents the description of some brain MR images. Figures 11.8 and 11.9 show some brain MR images, along with the segmented images obtained using the FMWCM method, index of area coverage (IOAC) [26], 1D fuzzy correlation (1-DFC) [27], 2D fuzzy correlation (2-DFC) [27], conditional entropy [21–23], and the method proposed by Otsu [20]. While Fig. 11.8 represents the results for I-629, I-647, I-677, I-704, and I-734, Fig. 11.9 depicts the results of I-760, I-761, I-763, I-768, and I-788. In Table 11.2, the details of these brain MR images are provided and Table 11.3 shows the values of the thresholds obtained using different methods. Unlike existing thresholding methods, the FMWCM scheme can detect multiple segments of the objects if there exists. All the results reported in this chapter clearly establish the fact that the FMWCM method is robust in segmenting brain MR images compared to existing thresholding methods. None of the existing thresholding

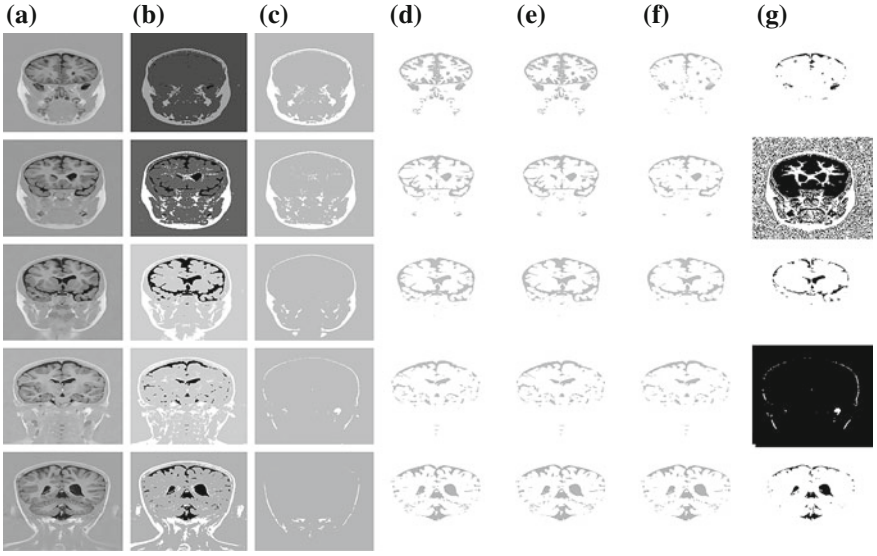

Fig. 11.9 Brain MR images (*top* to *bottom*: I-760, I-761, I-763, I-768, I-788) along with the segmented images **a** Image ; **b** FMWCM ; **c** IOAC ; **d** 1-DFC ;**e** 2-DFC ; **f** Entropy and **g** Otsu

Table 11.2 Description of some brain MR images

Image Index	Size ($M \times N$)	Gray Level	Gray value	
			Maximum	Minimum
I-629	256 × 256	115	114	0
I-647	256 × 256	515	514	0
I-677	512 × 512	375	374	0
I-704	640 × 448	1378	1377	0
I-734	512 × 360	730	729	0
I-760	512 × 360	557	2239	1683
I-761	512 × 360	540	2244	1705
I-763	512 × 360	509	2217	1709
I-768	512 × 360	501	2188	1688
I-788	512 × 360	593	2242	1650

methods could generate as consistently good segments as the FMWCM algorithm. Also, some of the existing methods have failed to detect the object regions.

Table 11.3 Threshold values for different algorithms

Image Index	Threshold values					
	FMWCM	Otsu	Entropy	1-DFC	2-DFC	IOAC
I-629	13, 62, 103	32	25	58	55	25
I-647	37, 81, 342	94	150	12	18	20
I-677	70, 216, 313	82	216	350	342	31
I-704	97, 297, 926	266	264	924	922	70
I-734	138, 642	207	240	226	238	197
I-760	1777, 2064	1842	1904	1958	1952	2065
I-761	1929, 2067	2045	1928	1965	1954	2070
I-763	1946, 2069	1822	1924	1959	1949	2104
I-768	1896, 2065	2134	1933	1948	1948	2120
I-788	1905, 2069	1841	1928	1935	1951	2154

11.5 Conclusion and Discussion

The problem of segmentation of brain MR images has been addressed in this chapter. The importance of histogram thresholding for MR image segmentation problem and a brief survey on different existing thresholding methods have also been reported, along with their merits and demerits. Finally, a robust thresholding technique based on the fuzzy set theory is presented for segmentation of brain MR images. The histogram threshold is determined according to the similarity between gray levels. The fuzzy framework is used to obtain a mathematical model of such a concept. The edge information of each pixel location is incorporated to modify the co-occurrence matrix. The threshold determined in this way avoids local minima. This characteristic represents an attractive property of the FMWCM method. From the experimental results, it is seen that the FMWCM algorithm produces segmented images more promising than do the conventional methods.

References

1. Bezdek JC (1981) Pattern recognition with Fuzzy objective function algorithm. Plenum Press, New York
2. Bezdek JC, Hall LO, Clarke LP (1993) Review of MR image segmentation techniques using pattern recognition. Med Phys 20(4):1033–1048
3. Bezdek JC, Pal SK (1992) Fuzzy models for pattern recognition: methods that search for structures in data. IEEE Press, New York
4. Brandt ME, Bohan TP, Kramer LA, Fletcher JM (1994) Estimation of CSF, white and gray matter volumes in hydrocephalic children using Fuzzy clustering of MR images. Comput Med Imaging Graph 18:25–34
5. Cagnoni S, Coppini G, Rucci M, Caramella D, Valli G (1993) Neural network segmentation of magnetic resonance spin echo images of the brain. Int J Biomed Eng 15(5):355–362

6. Hall LO, Bensaid AM, Clarke LP, Velthuizen RP, Silbiger MS, Bezdek JC (1992) A comparison of neural network and Fuzzy clustering techniques in segmenting magnetic resonance images of the brain. IEEE Trans Neural Network 3(5):672–682
7. Haralick R, Shapiro L (1985) Survey: image segmentation techniques. Comput Vision Graph Image Proc 29:100–132
8. Hassanien AE (2007) Fuzzy rough sets hybrid scheme for breast cancer detection. Image Vision Comput 25(2):172–183
9. Hassanien AE, Abraham A, Peters JF, Schaefer G, Henry C (2009) Rough sets and near sets in medical imaging: a review. IEEE Trans Inf Technol Biomed 13(6):955–968
10. Lee C, Hun S, Ketter TA, Unser M (1998) Unsupervised connectivity based thresholding segmentation of midsaggital brain MR images. Comput Biol Med 28:309–338
11. Leemput KV, Maes F, Vandermeulen D, Suetens P (1999) Automated model-based tissue classification of MR images of the brain. IEEE Trans Med Imaging 18(10):897–908
12. Li CL, Goldgof DB, Hall LO (1993) Knowledge-based classification and tissue labeling of MR images of human brain. IEEE Trans Med Imaging 12(4):740–750
13. Maji P, Kundu MK, Chanda B (2006) Segmentation of brain MR images using Fuzzy sets and modified co-occurrence matrix. In: Proceedings of the IET international conference on visual information, engineering, pp 327–332
14. Maji P, Kundu MK, Chanda B (2008) Second order Fuzzy measure and weighted co-occurrence matrix for segmentation of brain MR images. Fundamenta Informaticae 88(1–2):161–176
15. Maji P, Pal SK (2007) RFCM: a hybrid clustering algorithm using rough and Fuzzy sets. Fundamenta Informaticae 80(4):475–496
16. Maji P, Pal SK (2007) Rough set based generalized Fuzzy C-means algorithm and quantitative indices. IEEE Trans Syst Man Cybern Part B Cybern 37(6):1529–1540
17. Maji P, Pal SK (2008) Maximum class separability for rough-Fuzzy C-means based brain MR image segmentation. LNCS Trans Rough Sets 9:114–134
18. Manousakes IN, Undrill PE, Cameron GG (1998) Split and merge segmentation of magnetic resonance medical images: performance evaluation and extension to three dimensions. Comput Biomed Res 31(6):393–412
19. Mushrif MM, Ray AK (2008) Color image segmentation: rough-set theoretic approach. Pattern Recogn Lett 29(4):483–493
20. Otsu N (1979) A threshold selection method from gray level histogram. IEEE Trans Syst Man Cybern 9(1):62–66
21. Pal NR, Pal SK (1989) Entropic thresholding. Signal Process 16(2):97–108
22. Pal NR, Pal SK (1989) Object-background segmentation using new definitions of entropy. IEE Proc-E 136(4):284–295
23. Pal NR, Pal SK (1991) Entropy: a new definition and its applications. IEEE Trans Syst Man Cybern 21(5):1260–1270
24. Pal NR, Pal SK (1992) Higher order Fuzzy entropy and hybrid entropy of a set. Inf Sci 61:211–231
25. Pal NR, Pal SK (1993) A review on image segmentation techniques. Pattern Recogn 26(9):1277–1294
26. Pal SK, Ghosh A (1990) Index of area coverage of Fuzzy image subsets and object extraction. Pattern Recogn Lett 11(12):831–841
27. Pal SK, Ghosh A (1992) Image segmentation using Fuzzy correlation. Inf Sci 62(3):223–250
28. Pal SK, Ghosh A, Shankar BU (2000) Segmentation of remotely sensed images with Fuzzy thresholding, and quantitative evaluation. Int J Remote Sens 21(11):2269–2300
29. Pal SK, King RA, Hashim AA (1983) Automatic gray level thresholding through index of fuzziness and entropy. Pattern Recogn Lett 1:141–146
30. Pal SK, Mitra P (2002) Multispectral image segmentation using the rough set-initialized-EM algorithm. IEEE Trans Geosci Remote Sens 40(11):2495–2501
31. Rajapakse JC, Giedd JN, Rapoport JL (1997) Statistical approach to segmentation of single channel cerebral MR images. IEEE Trans Med Imaging 16:176–186
32. Rangayyan RM (2004) Biomedical image analysis. CRC Press, Boca Raton

33. Rosenfeld A, Kak AC (1982) Digital picture processing. Academic Press Inc., New York
34. Sahoo PK, Soltani S, Wong AKC, Chen YC (1988) A survey of thresholding techniques. Comput Vision Graph Image Process 41:233–260
35. Singleton HR, Pohost GM (1997) Automatic cardiac MR image segmentation using edge detection by tissue classification in pixel neighborhoods. Magn Reson Med 37(3):418–424
36. Suetens P (2002) Fundamentals of medical imaging. Cambridge University Press, Cambridge
37. Suzuki H, Toriwaki J (1991) Automatic segmentation of head MRI images by knowledge guided thresholding. Comput Med Imaging Graph 15(4):233–240
38. Tobias OJ, Seara R (2002) Image segmentation by histogram thresholding using Fuzzy sets. IEEE Trans Image Process 11(12):1457–1465
39. Tou JT, Gonzalez RC (1974) Pattern recognition principles. Addison-Wesley, Reading, MA
40. Wells WM III, Grimson WEL, Kikinis R, Jolesz FA (1996) Adaptive segmentation of MRI data. IEEE Trans Med Imaging 15(4):429–442
41. Weszka JS, Rosenfeld A (1979) Histogram modification for threshold selection. IEEE Trans Syst Man Cybern 9(1):38–52
42. Widz S, Revett K, Slezak D (2005) A hybrid approach to MR imaging segmentation using unsupervised clustering and approximate reducts. In: Proceedings of the 10th international conference on rough sets, Fuzzy sets, data mining, and granular, computing, pp 372–382
43. Widz S, Revett K, Slezak D (2005) A rough set-based magnetic resonance imaging partial volume detection system. In: Proceedings of the 1st international conference on pattern recognition and machine intelligence, pp 756–761
44. Widz S, Slezak D (2007) Approximation degrees in decision reduct-based MRI segmentation. In: Proceedings of the frontiers in the convergence of bioscience and information technologies, pp 431–436
45. Xiao K, Ho SH, Hassanien AE (2008) Automatic unsupervised segmentation methods for MRI based on modified Fuzzy C-means. Fundamenta Informaticae 87(3–4):465–481
46. Zadeh LA (1965) Fuzzy sets. Inf Control 8:338–353

About the Authors

Pradipta Maji received the BSc degree in Physics, the MSc degree in Electronics Science, and the PhD degree in the area of Computer Science from Jadavpur University, India, in 1998, 2000, and 2005, respectively.

Currently, he is an Associate Professor in the Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India. He is the Principal Investigator of the Biomedical Imaging and Bioinformatics Lab of the Institute. He was associated with the Center for Soft Computing Research: A National Facility, Indian Statistical Institute, Kolkata, India, from 2005 to 2009. During the period of September 2000 to April 2004, he was a Research Scholar of Department of Computer Science and Technology, Bengal Engineering College (currently known as Bengal Engineering & Science University), Shibpur, Howrah, India. From 2002 to 2004, he also served as a Research & Development Consultant of Cellular Automata Research Laboratory, Kolkata, India. His research interests include pattern recognition, machine learning, soft computing, computational biology and bioinformatics, medical image processing, and so forth. He has published more than 90 papers in international journals and conferences. He is an author of a book published by Wiley-IEEE Computer Society Press, and also a reviewer of many international journals.

Dr. Maji has received the 2006 Best Paper Award of the International Conference on Visual Information Engineering from The Institution of Engineering and Technology, U.K., the 2008 Microsoft Young Faculty Award from Microsoft Research Laboratory India Pvt., the 2009 Young Scientist Award from the National Academy of Sciences, India, and the 2011 Young Scientist Award from the Indian National Science Academy, India, and has been selected as the 2009 Young Associate of the Indian Academy of Sciences, India.

Sushmita Paul received the BSc degree in Biotechnology from University of Rajasthan, India in 2005, the MSc degree in Bioinformatics from Banasthali Vidyapith, Rajasthan, India in 2007, and the PhD degree in Biophysics, Molecular Biology and Bioinformatics from University of Calcutta, India in 2014.

Currently, she is a Visiting Scientist in the Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India. She is also a member of the Biomedical Imaging

P. Maji and S. Paul, *Scalable Pattern Recognition Algorithms*,
DOI: 10.1007/978-3-319-05630-2,

and Bioinformatics Lab of the Institute. She was a Visiting Research Fellow of the Center for Soft Computing Research: A National Facility, Indian Statistical Institute, Kolkata, India, from 2007 to 2008. During the period of June 2008 to March 2011, she was a Project Linked Person in the Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India. From April 2011 to December 2013, she was a Senior Research Fellow of Council of Scientific and Industrial Research (CSIR), Government of India. Her research interests include computational biology and bioinformatics, pattern recognition, soft computing, and so forth. She has published around 20 papers in international journals and conferences. She is also a reviewer of many international journals and conferences.

Dr. Paul has received the 2009 Best Paper Award of the International Conference on Information Technology from the Orissa Information Technology Society, India, the 2011 Senior Research Fellowship and the 2014 Research Associateship of the CSIR, Government of India.

Index

P. Maji and S. Paul, *Scalable Pattern Recognition Algorithms*,
DOI: 10.1007/978-3-319-05630-2,