COMPUTATIONAL INTELLIGENCE METHODS FOR GENE SELECTION AND CLASSIFICATION

A
Thesis
Submitted to

LOVELY
PROFESSIONAL
UNIVERSITY

For the award of
DOCTOR OF PHILOSOPHY (Ph.D)
in
Computer Science and Engineering

By
Divya
Registration Number: 41300074

Supervised By

Dr. Babita Pandey nee Shukla
Associate Professor

Co-Supervised By

Dr. Devendra K Pandey
Assistant Professor

LOVELY FACULTY OF TECHNOLOGY AND SCIENCES
LOVELY PROFESSIONAL UNIVERSITY
PUNJAB
2017

DECLARATION

I hereby declare that the thesis entitled "Computational intelligence methods for gene selection and classification" submitted by me for the Degree of Doctor of Philosophy in Computer Science and Engineering is the result of my original and independent research work carried out under the guidance of Dr. Babita Pandey, Associate Professor, Department of Computer Applications, Lovely Professional University, Punjab, and Dr. Devendra K Pandey, Assistant Professor, Department of Biosciences, Lovely Professional University, Punjab, and it has not been submitted for the award of any degree, diploma, associateship, fellowship of any University or Institution.

Place:
Date:

Signature of the Candidate

CERTIFICATE

This thesis entitled "Computational Intelligence Methods for Gene Selection and Classification" submitted by Divya Anand of Lovely Professional University is a record of bona fide research work done by her and it has not been submitted for the award of any degree, diploma, associateship, fellowship of any University/Institution.

Place:

Date:

Signature of the Guide

Signature of the Co-Guide

ACKNOWLEDGEMENT

Firstly, I would like to express my sincere gratitude to my advisors Dr. Babita Pandey and Dr. Devendra K Pandey for the continuous support of my Ph.D study and the related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having better advisors and mentors for my Ph.D study.

Besides my advisors, I would like to thank my mother for her encouragement and support. Your prayer for me was what sustained me thus far. My special thanks also go to Aman Singh, who motivated me throughout the time. Without his precious support it would not be possible to conduct this research.

ABSTRACT

With the advance of computational intelligence methods (CIMs) in the medical field, researchers are now able to select the most discriminating genes for the classification or diagnosis of various diseases. The DNA microarray technology enables us to examine the thousands of genes simultaneously in a single experiment. The number of publicly available samples is very less. The typical nature of gene expression data sets, i.e., a large number of gene expression values and a small number of samples leads to the misclassification of diseases and increased classification cost. It is observed that most of the genes in the data set are irrelevant, redundant and uninformative as they are not specific to the disease. So, the selection of informative genes from this huge number of genes will help us in the correct classification of disease which ultimately increases the classification accuracy and decreases the classification costs.

In chapter 1, the review of literature is given where the use of CIMs is shown in solving the problems in various fields of bioinformatics. The research papers reviewed are shown in the tabular form. In chapter 2, we provide all the basic concepts used in this thesis. It includes the basic biological background information and the problem statement, introduction to the gene expression data, microarray technology, neuromuscular disorder classification problem and the associated issues, publicly available neuromuscular disorder data sets, feature selection and its models, classification methods, model selection parameters and model validation techniques.

In Chapter 3, an unsupervised approach of feature selection is employed to reduce the issue of dimensionality reduction, which leads to the clustering of discriminating genes and classification of samples of binary class data sets. The holdout validation technique is employed to divide the data into training set and test set, i.e., the whole data are divided according to the rule of conventional validation, i.e., 70% training data and 30% test data. Two clustering methods, k-means and hierarchical clustering methods with two distance metrics, i.e., euclidean and cosine metrics are employed to cluster the important genes. Further three classification algorithms, namely linear discriminant analysis (LDA), quadratic discriminant analysis (QDA) and k nearest neighbor (KNN) are implemented to classify the samples. So, the nine intelligent integrated approaches are implemented: K-means-LDA, K-means-QDA, K-means-KNN,

euclidean distance metric based hierarchical clustering-LDA, euclidean distance metric based hierarchical clustering-QDA, euclidean distance metric based hierarchical clustering-KNN, cosine distance metric based hierarchical clustering-LDA, cosine distance metric based hierarchical clustering-QDA and cosine distance metric based hierarchical clustering-KNN. The facioscapulohumeral muscular dystrophy (FSHD) dataset taken to evaluate the performance of these intelligent integrated methods contains a total of 33,297 genes and 50 samples where 26 samples are affected by FSHD and the rest 24 samples are healthy samples. Amongst all the nine intelligent integrated methods, the cosine distance metric based hierarchical clustering algorithm-KNN has given the best performance measures. It is observed that it is difficult to access the relevance of features after selecting the features using the unsupervised approach of feature selection.

In chapter 4, the supervised approach of feature selection is employed to overcome the issues observed in selecting features using the unsupervised approach of feature selection. The whole dataset is divided into training set and test set using 5-fold cross validation technique. Here two filter models of supervised feature selection, i.e., t-test and entropy are employed to select the important genes from a large number of genes. Two classification methods, namely KNN and linear support vector machine (SVM) are implemented to classify the samples using only the important genes selected using the filter models of supervised feature selection methods. So, the four intelligent integrated approaches implemented are t-test-KNN, t-test-linear SVM, entropy-KNN and entropy-linear SVM. Two datasets are used to access the performance of these integrated methods, i.e., juvenile dermatomyositis (JDM) and FSHD. The JDM dataset contains 22,645 genes and 39 samples. From these samples, 21 samples are affected by JDM and the rest 18 samples are healthy samples. The FSHD dataset contains 22,645 genes and 32 samples. In this dataset, 14 samples are affected by FSHD and 18 samples are healthy samples. From the above mentioned four intelligent integrated methods, the integration of entropy with KNN has given the best performance measures. Here, it is observed that the features are selected without interacting with the classifiers. So, the filter models of supervised feature selection can be implemented as a preprocessing step in selecting the highly relevant and important genes.

In chapter 5, the preselection of genes is done by using the filter model of feature selection as a preprocessing step and the selection of genes is done using an embedded model of feature selection. Between the filter model preselection phase and classification phase, a new gene selection phase is embedded which selects out the most discriminating genes from a large number of genes. Here, to segregate the training set and test set out of the whole data set, leave-one-out cross-validation (LOOCV) technique is employed. In the gene preselection or pre-processing phase, a filter model, i.e., t-test is implemented to remove out redundant and noisy genes. In the gene selection phase, an embedded model, i.e., genetic algorithm (GA) is applied to select the most discriminating and important genes. Here, the fitness function in GA is evaluated using LDA, QDA and KNN one by one with varying number of genes. Hence the classification of samples using the only selected genes is done by using the same three classification algorithms, i.e., LDA, QDA and KNN. In each experiment the number of genes is varied and the performance measures are calculated. So, the intelligent integrated approaches in this methodology are t-test-GA-LDA, t-test-GA-QDA and t-test-GA-KNN. The performance of these integrated methods is accessed on the FSHD dataset of 33,297 genes and 50 samples. Here the integrated method t-test-GA-KNN has given the best classification accuracy, i.e., 100% with just 10 genes. The addition of embedded model in between the filter model and classification algorithm has enhanced its performance.

In chapter 6, the problem of gene selection for multi-class classification is resolved by implementing a novel intelligent integrated technique. The data sets are divided into training set and test set using 5-fold cross validation technique. The genes are preselected using the bhattacharyya coefficient of all the genes in the samples. The top-valued genes are chosen and given as input to the next step where the most discriminating genes are selected using GA. Here also, the fitness function is evaluated using different classification algorithms like LDA, QDA, KNN, linear SVM and RBF SVM one by one. Then the samples are classified using LDA, QDA, KNN, linear SVM and RBF SVM uses only the selected genes. The intelligent integrated approaches are bhattacharyya-GA-LDA, bhattacharyya-GA-QDA, bhattacharyya-GA-KNN, bhattacharyya-GA-linear SVM, bhattacharyya-GA-RBF SVM. These integrated approaches are implemented on two data sets. The first data set contains a total of 22,645 genes, 72 samples and 5 classes. The second data set contains 22,283 genes, 55 samples and 6 classes. Here the integration of bhattacharyya-GA with RBF SVM has given highest performance measures in both of the data sets.

In chapter 7, the challenge of feature selection is removed using the median matrix. The gene expression matrix is processed to create a median matrix for the selection of compact and different subsets of genes for every class. The classification algorithms use the combination of selected genes for prediction of the kind of neuromuscular disorder samples. The various classification algorithms employed are linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), k-nearest neighbor (KNN), linear support vector machine (Linear SVM) and radial basis function based support vector machine (RBF SVM). The classification algorithms use the “one-versus-all” approach to decompose the multi-class classification problem into binary class classification problem. The accuracy and effectiveness of the proposed model are exhibited through analysis of publicly available microarray data set of 13 neuromuscular disorders. It selects only a few biomarker and dissimilar genes for each class of neuromuscular disorder. It selects a minimum of 4 genes in one class and a maximum of 19 genes in another class. The integration of the proposed method of gene selection with RBF SVM classification algorithm has outperformed in most of the cases.

All the chapters are concluded in chapter 8. A future work of the present work is also given in this chapter.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CIM	Computational intelligence method
ANN	Artificial neural network
FL	Fuzzy logic
DM	Data mining
SI	Swarm intelligence
EA	Evolutionary algorithm
KBM	Knowledge based method
GE	Genomics
TR	Transcriptomics
PR	Proteomics
CBR	Case based reasoning
DNA	Deoxyribose nucleic acid
ANN-DM	Artificial neural network-data mining
ANN-EA	Artificial neural network-evolutionary algorithm
ANN-FL	Artificial neural network-fuzzy logic
ANN-PCA	Artificial neural network-principal component analysis
ANN-SI	Artificial neural network-swarm intelligence
CBR-DM	Case-based reasoning-data mining
DM-EA	Data mining-evolutionary algorithm
DM-LDA	Data mining- linear discriminant analysis
DM-NB	Data mining-naïve Bayes
DM-SI	Data mining-swarm intelligence

EA-FL	Evolutionary algorithm-fuzzy logic
EA-NB	Evolutionary algorithm-naïve Bayes
EA-SI	Evolutionary algorithm-swarm intelligence
ANN-Bayesian classifier	Artificial neural network-Bayesian classifier
ANN-DM-EA	Artificial neural network-data mining-evolutionary algorithm
ANN-DM-PCA	Artificial neural network-data mining- principal component analysis
ANN-CBR-DM	Artificial neural network-case-based reasoning-data mining
ANN-DM-RF	Artificial neural network-data mining-random forest
ANN-DM-FL	Artificial neural network-data mining-fuzzy logic
ANN-DM-SI	Artificial neural network-data mining-swarm intelligence
ANN-EA-FL	Artificial neural network-evolutionary algorithm-fuzzy logic
ANN-EA-PCA	Artificial neural network-evolutionary algorithm-principal component analysis
ANN-EA-RF	Artificial neural network-evolutionary algorithm-random forest
ANN-NB-RF	Artificial neural network-naïve Bayes-random forest
ANN-PCA-PLS	Artificial neural network-principal component analysis-partial least square
ANN-EA-FL-SI	Artificial neural network-evolutionary algorithm-fuzzy logic-swarm intelligence
AI	Artificial intelligence
BLAST	Basic local alignment search tool
BP	Back propagation
FF	Feed-forward
FB	Feed-backward
BRNN	Bidirectional recurrent neural network

IEBRNN	Interaction enriched bidirectional recurrent neural network
PNN	Probabilistic neural network
RBFNN	Radial basis function neural network
GRNN	Generalized regression radial basis neural network
BBFNN	Bio-basis function neural network
SVM	Support vector machine
RBF	Radial basis function
OVA	One-versus-all
OVO	One-versus-one
MLP	Multilayer perceptron
SOM	Self-organizing map
GCS	Growing cell structure
CV	Cross validation
LOOCV	Leave-one-out cross-validation
NCBI	National Centre for Biotechnology Information
EPD	Eukaryotic promoter database
RFE	Recursive feature elimination
DSSP	Database of secondary structure of proteins
PSP	Protein structure prediction
PSSP	Protein secondary structure prediction
CPM	Compound pyramid model
GNN	Gray neural network
FRAN	Fuzzy resource allocating network

GA	Genetic algorithm
PBIL	Population based incremental learning
ODB	Omics data bank
MOGAMOD	Multi-objective genetic algorithm for motif discovery
NSGA	Non-dominated sorting genetic algorithm
REGAL	RNA editing site prediction by genetic algorithm learning
MRMR	Minimum redundancy maximum relevance
ACO	Ant colony optimization
DT	Decision tree
KNN	K nearest neighbor
FIS Tree	Frequent Itemset Tree
BSC Tree	Bit string compression tree
NB	Naïve Bayes
RDA	Regularized discriminant analysis
LIBSVM	Library of support vector machine
PSO	Particle swarm optimization
ABC	Artificial bee colony
GSA	Gravitational search algorithm
AIS	Artificial immune system
MOABC	Multi-objective artificial bee colony
MOGSA	Multi-objective gravitational search algorithm
TRANSFAC	Transcription factor database
EPSO	Enhancement of particle swarm optimization

BPSO	Binary particle swarm optimization
IF-ABC	Internal feedback based on artificial bee colony
REDISC	Redundancy elimination based on discriminative contribution
ANOVA	Analysis of variance
SFAM	Simplified fuzzy art map
FCCEGS	Fuzzy C-Mean Clustering based Enhanced Gene Selection method
MOEA	Multi-objective evolutionary algorithm
SGERD	Steady-state genetic algorithm for extracting fuzzy classification rules from data
RFR	Recursive feature replacement
IVGA	Independent variable group analysis
GAGS	Genetic algorithm gene selection
MTSVSL	Multitask support vector sample learning
MTL	Multitask learning
GP	Genetic programming
ESOINN	Enhanced self-organized incremental neural network.
RNN	Recurrent neural network
DLD	Diagonal linear discriminant
SVD	Singular value decomposition

Chapter 1

Introduction

The computational intelligence is a branch of computer science which studies and solves the problems for which there is no effective computational algorithm. It improves the intellectual behavior of machines by incorporating the elements of learning, adaptation, heuristic and meta-heuristic optimization. The computational intelligence methods (CIMs) solve the complex problems by mimicking the characteristics of human beings like logic, understanding, reasoning, planning, learning, solving and self-organizing. These methods are widely applied in various domains, i.e., computer science, data analysis, optimization, medicine, business, banking, economics, forensic and security and manufacturing systems. In computer science, CIMs are extensively employed to resolve various challenges, namely the dimensionality reduction, classification and regression, optimization, data mining and clustering of data. The CIMs encompass of artificial neural network (ANN), evolutionary algorithm (EA), data mining (DM), fuzzy logic (FL), swarm intelligence (SI), and many others.

These methods have been broadly deployed for the dimensionality reduction of the disease data sets and their classification. Earlier the diseases were diagnosed or classified based on their appearances and the morphological properties. But the existing classes of a disease are found to be heterogeneous which follows a distinct pattern of pathogenesis. So for the accurate classification of these diseases, we need to consider the gene expression levels for which the changes in the activity of one or more genes lead to the occurrence of a disease. Recently the microarray technology came into the picture which analyzes the whole genome simultaneously. It has motivated the use of gene expression levels for the accurate diagnosis of diseases. But the foremost challenge arises in the diagnosis of a disease using the gene activity levels monitored through microarray technology is the high dimension of gene expression data sets of diseases. These data sets normally contain a huge number of genes which pose a problem in the accurate diagnosis. The next leading challenge comes in the way of diagnosis is the few numbers of publicly available samples. The numbers of samples available are very less as compared to the number of genes which leads to the problem of overfitting. Another issue is the existence of redundant, noisy and irrelevant genes in the data set. So, in order to remove out these genes from the data set, dimensionality reduction is usually performed so that the diseases are accurately classified. Various CIMs have been employed in the past for the selection of discriminating genes which helps in correct classification of diseases.

This chapter is structured as follows: Section 1.1 gives the review of literature which consists of the challenges in using computational intelligence methods in bioinformatics and the various methods to solve these challenges. Section 1.2 presents the individual knowledge based methods employed in bioinformatics for solving various tasks. Section 1.3 gives the individual computational intelligence methods deployed in bioinformatics for resolving various challenges. Section 1.4 lists all the integrated methods employed to solve the encountered challenges. Section 1.5 presents the results and a discussion of the methods deployed. Section 1.6 concludes the review of the literature conducted. Section 1.7 gives the motivation behind the present work. Section 1.8 details all the objectives to reach the goal. Section 1.9 presents the plan of the thesis and gives the outline and a brief description of each chapter. The detailed description is as follows.

1.1 Review of Literature

This study includes a literature review on the application of knowledge based methods (KBMs) and CIMs in the field of bioinformatics from year 1992 to 2016. The term “biological informatics” is often abbreviated as “bioinformatics”. It is a combination of biology and computer science that deals with the computational methods facilitating capturing, storing, organizing, analyzing, retrieving and interpreting the biological data [1]. It is applied to obtain the biological data, maintaining the vast variety of data in the databases, developing a method to incorporate the allied data from different sources and building up a way to extract the useful information from these databases. To accelerate and boost up the biological research, the KBMs and CIMs are widely deployed.

The three sub-fields of bioinformatics are reviewed, where KBMs and CIMs are employed. These are genomics (GE), transcriptomics (TR) and proteomics (PR). The branch GE involves the analysis of an organism’s nucleotide sequence. With the facilitation of GE, it is now feasible to estimate the number of genes in an organism. It identifies the cellular components such as proteins, rRNA, tRNA, etc., and analyzes the sequences attributed to the structural genes, regulatory sequences and non-coding sequences [2]. The branch of TR involves the study of mRNA molecules which deals with the monitoring of expression level of genes between different conditions and comparing the expression levels of diseased samples and control samples. The branch PR deals with the final product, i.e., protein and their interaction. It also involves the amino acids sequencing in a protein, predicting the structure and function of a protein [2].

KBMs and CIMs are highly deployed to solve the challenges encountered in the above mentioned branches. In the literature, KBMs are used to analyze and interpret the bioinformatics data [3] [4]. CIMs are widely deployed to select and extract the relevant biological knowledge from databases [5] [6], to envisage the uniqueness of biological systems and to present model to symbolize the biological knowledge. The first well-known application found in bioinformatics is based on distinguishing translation initiation sites in prokaryotic organisms [7], and since then, a number of applications are developed using KBMs and CIMs for dealing with wide range of challenges in GE, TR and PR.

1.1.1 Challenges in bioinformatics

The major challenges found in the literature in area of GE are interpreting genotypic drug resistance test to support the diagnosis of HIV, identification of intron-exon boundaries, normalization of cDNA microarray data, DNA sequencing and its analysis, prediction of DNA splice sites, alignment of nucleic acid sequences, genome wide identification of specific nucleotides, operons prediction, promoter recognition, genome sequence analysis, identification of gene regulatory networks, DNA motif discovery, optimization of multiple sequence alignment and phylogenetic inference.

The foremost problem or challenge noticed in the branch of TR is the microarray gene expression data classification. Because of the higher dimension of microarray data sets, the problem arises in the accurate classification. So, there is a need to reduce the dimension of these data sets. The procedure of classification of the microarray data set is by first reducing the dimension and then classifying the disease data set. This is done by first clustering/selection/extraction of genes and then classifying the data using only those genes. The first way is gene expression data clustering and classification of diseases. The second procedure is gene selection and disease classification, third is gene extraction and disease classification. Some researchers have addressed the complex issue of microarray gene expression classification in three phases, i.e., gene selection, extraction and classification.

The other challenges found in TR are the prediction of functionally related genes, gene expression ordering, identification and prediction of miRNA in viruses, miRNA classification; prediction of RNA predicting site, miRNA target prediction, prognosis and diagnosis of breast cancer. Some other tasks are to find the relationship between different genes, identification of genes of similar functions and mining co-regulated genes.

The problems encountered in the area of PR are protein function prediction, classification and prediction of β-turn types in proteins, protein peptide cleavage activity characterization, prediction of gene ontology functions of proteins, prediction of DNA binding domains in proteins, prediction of MHC class II peptide binding, prediction of functional association between proteins, signal peptide discrimination and cleavage site identification, protein and nucleic acid classification, feature selection in protein function prediction, protein secondary structure optimization, diagnosis of disease using serum proteomic profiling, protein and peptide classification, prediction and classification of protein coupled receptor, prediction of protein cellular localization sites and prediction of bacterial virulent proteins. But the major problem in the area of PR is protein structure prediction. Other problems are protein sequence classification, prediction of the HIV protease cleavage site in protein and protein fold recognition.

Some challenges are also found in other areas of bioinformatics like functional genomics and systems biology. In functional genomics, the tasks are prediction of protein-protein interaction, gene function prediction and functional analysis of gene expression data. The tasks of systems biology are detection of non-linear interactions among genes in common human diseases, extraction of association between biomarkers for cancer classification and reconstruction of gene regulatory network from gene expression.

1.1.2 Various methods to solve the challenges

Various individual and integrated KBMs and CIMs are applied to solve these bioinformatics challenges. KBM includes case based reasoning (CBR) which is a knowledge dominant method while CIMs incorporates ANN, EA, DM, FL, SI and many others which are data dominant methods. These methods offer complementary advantages and disadvantages, and when integrated, their advantages are explored and the disadvantages are mitigated. The integrated methods found in literature are artificial neural network-swarm intelligence (ANN-SI), artificial neural network-evolutionary algorithm-random forest (ANN-EA-RF), data mining-linear discriminant analysis (DM-LDA), artificial neural network-principal component analysis-partial least square (ANN-PCA-PLS), artificial neural network-data mining-random forest (ANN-DM-RF), artificial neural network-data mining-fuzzy logic (ANN-DM-FL), artificial neural network-evolutionary algorithm-fuzzy logic-swarm intelligence (ANN-EA-FL-SI), artificial neural network-evolutionary algorithm-fuzzy logic (ANN-EA-FL), artificial neural network-fuzzy logic (ANN-FL), case-based reasoning-data mining (CBR-DM), artificial neural network-principal component analysis (ANN-PCA), data mining-swarm intelligence (DM-SI), evolutionary algorithm-fuzzy logic (EA-FL), artificial neural network-data mining (ANN-DM), artificial neural network-evolutionary algorithm (ANN-EA), artificial neural network-Bayesian classifier (ANN-Bayesian classifier), data mining-evolutionary algorithm (DM-EA), evolutionary algorithm-swarm intelligence (EA-SI), artificial neural network-case-based reasoning-data mining (ANN-CBR-DM), artificial neural network-data mining-evolutionary algorithm (ANN-DM-EA), artificial neural network-data mining- principal component analysis (ANN-DM-PCA) and artificial neural network-evolutionary algorithm-principal component analysis (ANN-EA-PCA). The review is done on all the challenges found in the different fields of bioinformatics employing KBMs and CIMs. Thus, this will be helpful to the novice researchers in the computer science field to choose the accurate and efficient KBMs, CIMs and the integrated method to solve the problems encountered in these areas.

1.2 Individual Knowledge Based Methods in Bioinformatics

KBMs are a set of programs that support explicit representation of knowledge on a specific domain of expertise and use it to provide the solutions to problems in that domain [8]. KBMs are knowledge dominant and use rules, semantic nets, frames, scripts, etc., for knowledge representation. It can also employ artificial intelligence (AI) methods for problem-solving procedures to carry out human decision making, learning and action [3]. KBMs consist of software programs for knowledge acquisition, knowledge representation and inference engine.

1.2.1 Case based reasoning

In 1977, Roger Schank introduced the concept of CBR [9]. It is also known as 'Reasoning by remembering'. It is a problem-solving model in which the problems are resolved by storing, retrieving and adapting the solutions of previously encountered problems. It is based on the assumption that the problems of same kind tend to reappear and similar problems have similar solutions. The main component in CBR is a 'case'. A case represents the knowledge collected on previously experienced situations and is composed of three components: problem description, solution and the final state. The information stored in a case must be specific so that it can be used in future. A CBR cycle is composed of four phases which includes retrieve, reuse, revise and retain. In retrieve phase, according to the target problem, regain the most similar and appropriate cases from the case base. In reuse phase, adapt the solutions of the retrieved cases. In revise phase, alter the proposed solution, if necessary. In retain phase, after the successful adaptation, save the necessary information in the memory for future use [10]. The summary of CBR-based bioinformatics systems is given in table 1.1.

Table 1.1: CBR based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
GE	Montani et al. [4]	**Case**: Sequence alignment using BLAST. **Qualitative value Conversion**: Temporal abstraction techniques. **Retrieval**: Flexible retrieval which makes use of multi-dimensional orthogonal index structures.	Genome sequence analysis	GMOD database Chado

Notes: BLAST, basic local alignment search tool.

In case of CBR, it needs much less domain knowledge than the statistical-based and rule-based methods. It possesses various advantages like naturalness, modularity, applicability, easy knowledge acquisition, self-updatability, learning from experiences, ability to express specialized knowledge, reflection of human reasoning, handling unexpected or missing value and inference efficiency [11]. Generally CBR is applied in designing, planning, diagnosis, therapy, analysis and explanation.

1.3 Individual Computational Intelligence Methods in Bioinformatics

The formulation and application of CIMs have gained importance over the past few years. Nowadays, these are commonly applied in the various fields like business, science, medicine, etc. The CIMs includes methods like ANN, EA, DM, FL, SI and many others.

1.3.1 Artificial neural network

ANN is a biologically inspired structure of computational elements known as neurons. It consists of many neurons interconnected using connection links and is arranged in the form of layers. Various common advantages of ANN are high generalization power, graceful degradation ability, strong learning and non-linear transformation ability [1], high pattern recognition and data organization capabilities, robustness, reliability, high parallelism, high noise tolerance power, stronger fault tolerance ability and non-linear flexibility and non-linear function capability. The process of learning in ANN from its environment and advancing its performance through learning is the most crucial part. The various types of learning in ANN are: supervised, reinforcement and unsupervised learning. In supervised learning, the network is supplied with the output for every input pattern. In reinforcement learning, the network is supplied with only some analysis of the output of the input patterns, but not the correct outputs. In unsupervised learning, the network is not supplied with any output concerned with any input pattern.

In the literature of ANN, in both individual and integrated methods various configurations are found. The most widely employed algorithm for training is back propagation (BP) [12]–[34]. Based on the architectures of connection pattern, ANN is categorized into two types, namely feed-forward (FF) and feed-backward (FB) networks. The applications of FF networks are found in [12], [13], [16], [18], [19], [21], [23], [25], [28]–[33], [35]–[42]. The different types of activation functions applied in literature are linear function [35], logarithmic function [14], logistic function [15], tan-sig [42], non-linear function [22], gaussian function [43] [44], , gradient descent [18] [37] [45] [46] [47] and sigmoid function [14] [16] [25] [30] [35] [41] [48] [49].

Bidirectional recurrent neural network (BRNN) captures local dependencies without any prior defined promise about the size of sliding window. It is capable of capturing at least partial long-ranged information without overfitting [23]. Interaction-enriched bidirectional recurrent neural network (IEBRNN) is an enlarged form of BRNN, which operates on undirected graphs whose vertices are consecutively ordered. It can effectively exploit relational information [25].

Probabilistic neural networks (PNN) lies in the category of radial basis function neural network (RBFNN), which depends up on the Bayes' decision strategy and Parzen's method of density estimation [36]. PNN does not support heuristic search, and can tolerate erroneous samples and outliers. It is less sensitive to noise, able to handle all the asymmetrical misclassification costs, deal with large amount of data, have high generalization ability and do not need to reconfigure or retrain from the scratch when new training data is available but they require large amount of memory [36]. Generalized regression radial basis neural network (GRNN) is a type of PNN, which is capable of dealing with sparse and non-stationary data. It is able to converge with only few training samples available. Bio-basis function neural network (BBFNN) is made by replacing the radial basis function of RBFNN with bio-basis function. It requires less number of parameters and able to regulate the biological information, and is considered fast and robust [50].

Support vector machine (SVM) is the most widely employed method of ANN for the classification of data sets. It searches for a unique separating hyperplane that lies between two classes in the input space which maximizes the margin between the hyperplane and classes [51]. It provides better accuracy rate than other machine learning methods [48] [52]. SVM incorporates various kernels namely linear [16] [51] [53]–[59], polynomial [60] [61] and radial basis function (RBF) [41], [52], [58], [62]–[69]. SVM supports only binary class classification of the datasets. But the task of multi-class classification is also solved using SVM employing two approaches namely one-versus-all (OVA) and one-versus-one (OVO) [52] [60] [63] [70]–[72]. SVM also offers assorted advantages like ability to condense information, absence of local minima, good generalization capacity, high prediction ability, scalability, fast convergence, reliable prediction and robustness in the noisy environment, high accuracy, specificity and sensitivity and ability to handle high-dimensional data [51] [53] [58]. But it does not allow for knowledge extraction and automatic feature selection and provides low comprehensibility [53]. So, to handle the crucial issue like feature selection for classification, we need to first reduce the dimension of the data then use it for the classification tasks.

Multilayer perceptron (MLP) is also a type of ANN which belongs to the class of FF neural network with varying hidden layers and it requires preprocessing to avoid the overfitting of data [48]. A self-organizing map (SOM) is another admired neural network model which belongs to the category of competitive learning networks and it is a self-organizing and self-adaptive model [73]. It also possess advantages like low-dimensional topology and stable evolving properties [73]. But the various problems associated with SOM are the use of predefined topology, time-dependence of number of parameters [74], lack of interpretability and model selection. Growing cell structure (GCS) is a variation of SOM, which offers various advantages over SOM like self-adaptive topology, capability to achieve problem-dependent error measures, ability to handle both small and high-dimensional data, correct evaluation of probability densities of input signals and not necessitate to a-priori define the time-dependent or decay schedule parameters [74]. It is an efficient, user-friendly, effective and inexpensive option to support diagnostic tasks [74]. The summary of various applications of different ANN methods is shown in table 1.2.

Table 1.2**:** ANN based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
GE	Fu [12]	**Architecture**: Two layered FF network, fully connected architecture with one hidden layer and standard BP algorithm. **Activation function**: Based on the certainty factor model of MYCIN-like expert systems. **Weights**: -1 to 1.	DNA sequence analysis	Primate splice junction gene sequence data set and human gene data set

		Performance evaluation: 2-fold CV technique.		
GE	Beerenwinkel et al. [53]	**Classification**: C4.5 and SVM. C4.5, a heuristic divide and conquer strategy, reduced error pruning. SVM, linear kernel, Langrangian dual, Joachim's SVM. **Performance evaluation**: LOOCV technique.	Interpreting genotypic drug resistance tests to support the diagnosis of HIV	Clinical samples data set and first 220 to 250 amino acids
GE	Yoshihara et al. [75]	**Architecture**: A multimodal neural network composed of a multilayer neural network and decision module.	Identification of intron-exon boundaries	NCBI
GE	Deng et al. [35]	Non-linear normalization method, **Architecture**: 3 layered FF network. **Transfer function**: Output layer: Linear transfer function. Hidden layer: Sigmoid transfer function.	Normalization of complementary DNA microarray data	Gene data set of immune system diseases
GE	Frias et al. [13]	**Architecture**: MLP, FF network, 8-1-1 architecture, BP algorithm.	Promoter recognition	EPD and GenBank
GE	Zhang et al. [51]	**Data encoding**: Sparse encoding. **Projection of data**: Bayes Kernel, posterior probability, positive and negative encodings.	Prediction of DNA splice sites	Nucleotide sequences of splice site

		Classification: Linear SVM.		
GE	Liu et al. [14]	**Prediction of oligo specificity**: **Architecture**: BP algorithm, 17-4-1, 17-10-1, 17-16-1 and 17-22-1 architectures. **Training algorithm**: Batch mode training algorithm. **Activation function:** Input and output layer: Sigmoid activation function. Hidden layer: Logarithmic function. **Verification**: BLAST.	Genome wide identification of specific oligonucleotide	Human gene index, unique marker database and rat gene index databases
GE	Knott et al. [48]	**Architecture**: 3 layered architecture. **Transfer function**: Tangent sigmoid transfer function and Bayesian regularization.	Identification of gene regulatory networks	Hippocampus development dataset and artificial data set
TR	Guyon et al. [54]	**Feature ranking**: RFE, using the weight magnitude as a ranking criterion. **Classification**: Linear SVM, training using a soft-margin algorithm. **Performance evaluation**: LOOCV technique.	Gene selection and cancer classification	Leukemia and colon cancer data sets
TR	Berrar et al. [36]	**Probabilistic neural network**, a kind of RBFNN, Bayes' decision strategy and density estimation using Parzen's method.	Multiclass cancer classification	Leukemia cancer and NC160 multi-class data set

		Architecture: FF network with 2-1-1-1 architecture. **Performance evaluation**: Lift based scoring system and LOOCV technique.		
TR	Gomes et al. [43]	**SOM,** **Synaptic weight initialization**: At initial stage, no-of-genes/ 4 neurons. **Competitive process**: Sequentially presented vectors to NN, the minimum distance neuron from the input vector wins. **Cooperative process**: 1-dimensional neighborhood, Gaussian neighborhood activation function. **Adaptive process**: Winning neurons gives the synaptic weight updates and its neighborhood.	Gene expression ordering or rearrangement of gene expression data	DLBCL and S. Cerevisiae data sets
TR	Xu and Zhang [76]	**Gene selection**: Virtual gene, linear combination of genes. **Classification**: SVM.	Gene selection and cancer classification	Colon, leukemia and multi-class cancer data sets
TR	Ahmed [15]	**Architecture**: FFNN with BP, 2 to 15 hidden units. **Activation function**: Logistic activation function. **Architecture**: FFNN, 2 hidden	Diagnosis and survival prediction in colon cancer	Colorectal cancer data set and National Cancer database

		layers, 1 output layer, sensitivity analysis method, and standard second order conjugate gradient descent method.		
TR	Zhang and Deng [55]	**Gene preselection**: Family wise error rate. **Gene selection**: Bhattacharyya distance and sequential forward selection algorithm. **Classification**: Linear SVM.	Gene selection and cancer classification	Colon, DLBCL, leukemia, prostate and lymphoma cancer data sets
TR	Yendrapalli et al. [77]	**Gene selection**: T-test. **Classification**: Biased SVM and LIBSVM algorithm. **Performance evaluation**: LOOCV technique.	Gene selection and cancer classification	Leukemia, lymphoma, colon and prostate cancer data sets
TR	Chen and Lin [16]	**Sampling significant samples**: Support vector sampling technique. **Gene selection**: Signal to noise ratio. **Classification**: SVM and BPNN. SVM: Binary SVM, LIBSVM algorithm, linear kernel function, simple dot-product kernel. BPNN: Multilayered FF, BP neural network, positive propagation, error correction learning rule, tan-sigmoid	Gene selection and cancer classification	Leukemia and prostate cancer data sets

		transfer function, log-sigmoid transfer function. **Performance evaluation**: K-fold CV and LOOCV techniques.		
TR	Zheng and Liu [56]	**Gene selection**: Least absolute shrinkage and selection operator, Dantiz selector. **Classification**: Linear regression, linear SVM, logistic regression. **Performance evaluation**: 10-fold CV technique.	Gene selection and cancer classification	DLBCL, leukemia, prostate, colon and lymphoma cancer data sets
TR	Sahu et al. [44]	**Gene selection**: F-Score. **Gene extraction**: Autoregressive model and parameters are computed by Levinson Durbin's Recursive process. **Classification**: RBFNN, Gaussian function, stochastic gradient approach. **Performance evaluation**: LOOCV technique.	Gene selection, extraction and classification	Leukemia, colon, prostate, lymphoma and SRBCT datasets
TR	Ding et al. [70]	**Feature extraction**: n-gram. **Classification**: Multiclass SVM, OVA strategy.	miRNA classification	miRBase, SNORA2 and SNORA33 data sets
TR	Chen [71]	**Feature ranking**: Univariate feature ranking, RFE. **Classification**: Cumulative	Cancer stage classification	Bladder, prostate, cervical, lung,

		logit model, support vector ordinal regression, rank SVM, multiclass SVM, OVA approach, Weston-Watkins and Crammer-Singer strategies.		ovarian cancer data sets
TR	Gupta et al. [17]	**Architecture**: MLP, 3 layered architecture and BP algorithm.	Identification and prediction of miRNA in viruses	miRBase
TR	Arunkumar and Ramakrishnan [57]	**Normalization**: Min-max normalization. **Gene extraction**: T-test and absolute scoring. **Classification**: Linear SVM, proximal SVM and Newton SVM. **Performance evaluation**: K-fold CV technique.	Gene extraction and classification	Princeton microarray database
PR	Qian and Sejnowski [18]	**Architecture**: FF network, BP learning algorithm and gradient descent algorithm.	PSSP	BPD
PR	Wu et al. [19]	**Protein acid classification**: FF network, BP with 462-200-164, 462-200-180, 462-200-192,462-200-154 unit architectures. **Nucleic acid classification**: FF network, BP with 1088-50-28 unit architecture.	Protein and nucleic acid classification	PIR
PR	Rost and Sander [37]	**Architecture**: 2 layered FF network, sigmoidal trigger function, and gradient descent method.	PSSP	PDB, HSSP

PR	Zhang [20]	**Architecture**: MLP, BP algorithm.	PSSP	BPD
PR	Rost et al. [38]	**Architecture**: 3-levelled and 2-layered FF neural network architecture.	PSSP	PDB
PR	Chandonia and Karplus [39]	**Architecture**: FF network, fully connected and 399-2-2 unit architecture.	PSSP and structural class prediction	Globular protein data set
PR	Wu et al. [21]	**Sequence encoding**: n-gram hashing function. **Size reduction**: SVD, Latent semantic indexing approach, simple vector Lanczos method. **Classification**: 3 layered architecture, FF networks, BP and supervised learning algorithm.	Full-scale protein sequence classification	PIR
PR	Cai and Chou [22]	**Architecture**: BP model, multi-layered sensory structure with 160-1-8 unit architecture. **Activation function**: Non-linear activation function. **Learning:** Iterative self-learning.	Prediction of HIV protease cleavage sites in proteins	Oligopeptide data sets
PR	Cai et al. [73]	**SOM,** **Architecture**: Input units: 80, output nodes form a 3014*2 lattice, weight initialization with random values.	Classification and prediction of β-turn types in proteins	BPD
PR	Hua and Sun [62]	SVM, RBF kernel.	PSSP	RS126
PR	Baldi and Pollastri [23]	**Architecture**: BRNN, probabilistic graphical model,	Protein structure and function	PDB

		use of both forward and backward Markov chains of hidden states. FF network, nonlinear state transition functions, supervised training and a generalized form of gradient descent or BP through time and 1-4-1 plane architecture.	prediction	
PR	Nguyen and Rajapakse [72]	SVM, OVA and OVO approach, direct acyclic graph.	PSSP	RS126
PR	Zhu et al. [24]	**Architecture**: BP algorithm, 315-15-1 and 45-15-3 unit architectures.	PSSP	Homology derived structures of the protein data bank
PR	Nakayama et al. [78]	**Architecture**: Multimodal neural network, single multilayer neural network.	PSSP	HSSP databank
PR	Chen and Chaudhari [45]	**Architecture**: Bi-directional segmented-memory recurrent neural network, extension of real time recurrent learning algorithm, forward and backward propagations, gradient based learning.	PSSP	RS126
PR	Hu et al. [63]	**Architecture**: RBF kernel with OVO and OVA classifiers.	PSSP	RS126
PR	Wang et al. [64]	**Architecture**: Soft margin SVM, RBF kernel, LIBSVM algorithm.	PSSP	CB513 and RS126
PR	Ceroni et al. [25]	**Architecture**: IEBRNN, sequential supervised learning,	PSSP	PDB

		forward and backward state transition functions, two directed graphs, FF neural network with sigmoidal outputs and no internal hidden layer, maximum likelihood approach, BP algorithm.		
PR	Yang et al. [50]	**Architecture**: BBFNN, linear classifier, 3 layered architecture, pseudo inverse method, Bayes Rule, log-sensitivity index as stopping criterion.	Protein peptide cleavage activity characterization	Trypsin Cleavage and Factor Xa Cleavage data sets
PR	Bi et al. [65]	SVM, LIBSVM algorithm, RBF kernel.	Prediction of gene ontology functions of proteins	SWISS-PROT database, human proteome data set
PR	Nguyen and Rajapakse [60]	Multi-class SVM, gaussian kernel, linear kernel and polynomial kernel functions.	PSSP	RS126, CB396, CASP4, EVA and PSIPRED data sets
PR	Fei and Lusheng [52]	Binary classifier, soft margin SVM, multiple OVA classifiers, voting strategy, RBF kernel.	Prediction of DNA binding domains in proteins	PFam data sets, UniqueProt
PR	Wang and Li [26]	Profile encoder, **Architecture**: 3 layered and temporal hierarchical network architecture, BP algorithm, adaptive adjustment strategy, and conjugate gradient method.	PSSP	DSSP and PDB
PR	Kakumani et al. [27]	**Architecture**: Fully connected, MLP neural network, the BP algorithm.	PSSP	RS126

PR	Nielsen and Lund [28]	**Architecture**: Conventional FF network, gradient descent BP algorithm, 2, 10, 20, 40 and 60 hidden neurons.	Prediction of MHC class II peptide binding	IEDB HLA-DR, SYFPEITHI, Wang, Lin, IEDB EI-Manzalawy-UPDS, SRDS1, SRDS2 data sets
PR	Tang et al. [79]	Large margin method: multi-class separation margin.	PSSP	CB513, RS126
PR	Lin and Xiao [29]	**Architecture**: GNN, 2 layered architecture, FF network, BP algorithm. First layer: tangent sigmoid neurons, second layer: one pureline neuron.	PSSP	Uniprot
PR	Bidargaddi et al. [30]	**Architecture**: 21-13-3 unit architecture, FF network and BP algorithm. **Activation function**: Tan-sigmoid and log sigmoid activation functions. Constructive layer algorithm, scale conjugate gradient algorithm. Hidden Markov-Bayesian Segmentation: Segmental semi-markov model. Viterbi algorithm and forward-backward algorithm.	PSSP	PDB_SELECT, CB513, PDB data sets
PR	Thalatam et al. [49]	ANN.	PSSP	NCBI
PR	Mathkour and	Java object oriented neural	PSP	Microbial

	Ahmad [31]	network, **Architecture:** 4 layered architecture, FF network and BP algorithm.		genome of 2000 to 5000 genes
PR	Kim et al. [32]	**Architecture:** MLP, FF network, 3 layered architecture, simple weight decay technique and gradient descent BP technique.	Prediction of HIV-1 protease cleavage site in proteins	Artificial data set: Corral, Monk1, Monk3, Corral-100 and XOR-100
PR	Qu et al. [80]	**Encoding**: PSI-BLAST, PSI-Search, HMMER3 and AMPS. **Architecture**: CPM: 3 layered architecture. Comprehensive layer: multi-modal BP neural network, mixed-modal SVM. Kernel Judgment layer: KDD process and M algorithm, structural association classifier. Assistant Judgment layer: Attribute association classifier.	PSSP	RCASP256, RS126, CB513 and ASP256 data sets
PR	Priyadarshini et al. [33]	**Architecture**: Fully connected, FF network, BP algorithm, 3 layered architecture, gradient descent method. **Activation function**: Log sigmoidal function.	PSSP	PDB
PR	Florido et al. [34]	**Architecture**: MLP, interconnected processing neurons, BP algorithm, fully connected and 3 layered architecture.	Prediction of functional association between proteins	Yeast proteins from RefSeq database and Saccharomyces Cerevasive,

				Baker’s Yeast, STRING, SGD and blastp
PR	Liu et al. [40]	**Architecture**: Multilayered FF network with 3 hidden layers.	PSP	CB513 and RS126 data sets
PR	Wang et al. [46]	**Architecture**: 3 layered architecture, BPNN, gradient descent type BP algorithm.	Prediction of protease cleavage site of protein antigen	Antigen
PR	Abbasi et al. [81]	KNN, MLP, RBF and FRAN.	Protein fold recognition	PDB-40D
PR	Kazemian et al. [41]	SVM: RBF Kernel. ANN: 2 layered architecture, fully connected, FF network, sigmoid function, regression based NN training.	Signal peptide discrimination and cleavage site identification	UniProt
PR and TR	Wang et al. [82]	BAN: LN, NLN. LN: Modeled as neural network, fully connected, use of energy function. NLN: Non-linear BAN, addition of the sigmoidal transformation unit, winner takes all competition mechanism.	Extraction of association between biomarkers for cancer classification	Protein expression data set, Nasopharyngeal carcinoma, leukemia, colon and breast gene expression data sets
Functional GE	Urquiza et al. [58]	**Feature selection**: Margin based criteria, linear, sigmoid and zero-one utility functions. SVM, linear and RBF kernel, negative log likelihood function, multi-class classification methodology.	Prediction of protein-protein interaction	Yeast extracted from SwissPfam, GOA, MIPS, 3did, Hintdb

Notes: FF, feed-forward; BP, back propagation; CV, cross validation; SVM, support vector machine; LOOCV, leave-one-out cross-validation; NCBI, national center for biotechnology information; MLP, multi-layer perceptron; EPD, Eukaryotic promoter database; BLAST, basic local alignment search tool; RFE, recursive feature elimination; RBFNN, radial basis function neural network; SOM, self organizing map; FFNN, feed forward neural network; DLBCL, diffuse large B-cell lymphoma; LIBSVM, library of support vector machine; BPNN, back propagation neural network; OVA, one-versus-all; PSSP, protein secondary structure prediction; PIR, protein information resource; HIV, human immunodeficiency virus; BRNN, bidirectional recurrent neural network; PDB, protein data bank; IEBRNN, interaction enriched bidirectional recurrent neural network; BBFNN, bio-basis function neural network; DSSP, database of secondary structure of proteins; GNN, gray neural network; CPM, compound pyramid model; FRAN, Fuzzy resource allocating network.

1.3.2 Evolutionary algorithm

EAs are meta-heuristic algorithms which mimic the natural evolution and are used to find the approximate solutions. It includes the genetic algorithm (GA), population based incremental learning (PBIL) and many others. GA is an iterative algorithm which keeps on creating the new population. The main component in GA is a chromosome. A chromosome is the genetic representation of the possible key to the problem. The chromosomes of the initial population are chosen. The fitness value of the solution is compared to other solutions. Reproduction is the development phase in which the individuals are evaluated as per their fitness function using survival of the fittest technique. The operators employed in GA are the selection, the crossover and the mutation. To select the best chromosomes from each population for evolution to the next generation is called the selection operation. The next operator, the crossover operator is employed to produce two new “offspring” solutions from two “parent” solutions. In mutation, one single parent is chosen and a mutation is done. GA terminates when the highest number of iterations or some fitness criterion is reached. The summary of the application of EAs in bioinformatics is given in table 1.3.

GA is considered as a robust and efficient technique in excluding redundant features [88]. It is capable of classification of linked pathways and is efficient in avoiding highly accessible regions in the energy landscape increasing the probability of reaching the global minimum [93]. We do not necessitate need any preceding knowledge regarding the search space. It also possesses various advantages like slower convergence rate, multiple search points, and the ability to escape from local optima. It includes interaction and correlation between features and avoids the issue of overfitting. Along with these advantages, the various disadvantages of GA are high complexity [86], large memory storage, poor hill climbing capability and high computation time [86] [89].

Table 1.3: EA based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
GE	Blazewicz et al. [83]	Chromosome: Permutation of indices of the oligonucleotide from the spectrum; adjacency based encoding. Population: Randomly generated according to the uniform distribution. Selection: Stochastic remainder method without replacement. Crossover: Greedy crossover. Termination: 20 iterations.	DNA sequencing	GenBank
GE	Wang et al. [66]	Multi approach guided GA, Chromosome: String of Operons. Population: 20 Chromosomes. Selection: Classical roulette wheel selection. Crossover: Classical single point crossover, rate: 0.33. Mutation: 2 steps mutation process, rate: 0.02.	Operons prediction	GenBank, RegulonDB and ODB
GE	Kaya [84]	MOGAMOD, NSGA-II algorithm. Three objectives: Support maximization, motif length and similarity.	Discovery of optimal motifs from sequence data	TRANSFAC

		Chromosome: Initial location of a potential motif on all the target sequences. Population: 300 chromosomes. Selection: Probabilistic selection. Crossover: One-point crossover, non-uniform arithmetical crossover method, probability: 0.8. Mutation: 3 mutational operators and generalization by genetic operators, rate: 0.3.		
GE	Vijayvargiya and Shukla [85]	Niched Pareto GA, Objectives: Motif length and consensus similarity score. Chromosome: Individual is represented by position based representation approach; numerical encoding. Population: Generated using multiple attribute representation. Selection: Pareto domination tournament selection. Crossover: One-point crossover. Mutation: Mutation of randomly selected victim individual motif. Insertion and Evaluation: Motif length and consensus simulating. Termination: Stagnation and till the specified number of generations.	Identification of variable length regulatory motifs	Synthetic data set and promoter sequence data set of S. Cerevisiae
GE	Ortuno et al. [86]	Based on NSGA-II, Methodologies used: ClustalW, MUSCLE, Kalign, Mafft,	Optimization of multiple	BaliBASE

		RetAlign, TCOFFEE, ProbCons and FSA. Chromosome: Multiple sequences encoded as a matrix of real integer numbers. Population: Alignments obtained from 8 methodologies; 100 chromosomes. Selection: Pareto fronts. Crossover: Probability: 0.8. Mutation: Only gaps are mutated in order to maintain the position of amino acids; probability: 0.2. Termination: 200 generations.	sequence alignment	
TR	Gesu et al. [87]	GenClust, Chromosome: 32 bit string; binary encoding. Population: n chromosomes arranged in any order. Selection: Elimination of duplicates by keeping only the rightmost string. Crossover: Standard one-point crossover with probability: 0.9. Mutation: 1-bit and silent mutation with probability: 0.01. Termination: 500 iterations.	Gene expression data clustering	RCNS, YCC, RYCC, PBM, RPBM data sets
TR	Thompson and Gopal [88]	REGAL, Chromosome: Binary encoding. Population: 50 chromosomes. Replacement: Rank replacement.	Prediction of RNA predicting site	GenBank

		Crossover: Single-point crossover. Mutation: Probabilistic mutation. Termination: 300 generations.		
TR	To and Vohradsky [89]	Parallel GA, Chromosome: Real number and value encoding. Population: 1000 chromosomes. Selection: Chromosomes with the least value of fitness function. Reproduction: Probability: 0.1. Crossover: Probability: 0.9. Termination: 500 generations. For parallel scheme, Island model Topology: Ring topology, Migration rate: 5-10%. Migration frequency: After 10 generations. Sub-population sizes: 500, 260.	In a gene expression data, find genes of similar functionality	S. Coelicolor artificial random data sets
TR	Perez et al. [90]	**PBIL;** Chromosome: Set of genes, binary encoding. Population: 1000 chromosomes. Mutation: Probability: 0.02, mutation shifts: 0.05. Termination: 100 generations. Learning rate: 0.1.	Feature extraction	3-class leukemia data set

		Negative learning rate: 0.1.		
PR	Nemati et al. [91]	GA and ACO run in parallel. GA, Population: 50 chromosomes. Crossover: Probability: 0.7. Mutation: Probability: 0.005. Iterations: 100. ACO, Population: 50 chromosomes. Initial Pheromone: 1. Importance of pheromone level: 1. Importance of heuristic information: 0.1.	Feature selection in protein function prediction	GPCR-PROSITE and ENZYME-PROSITE data sets from UniProt and Prosite database
PR	Su et al. [92]	Elite based reproduction strategy-Genetic algorithm: Initialization: Candidate conformation in 2D triangular lattice. Population: Randomly generated; 200 chromosomes. Reproduction: Elite based reproduction strategy. Crossover: Two-point crossover, rate: 0.8. Mutation: Uniform mutation, rate: 0.4. Local Search: 2. Termination: 200 generations. Enhancement of exploitation capability: Hill climbing.	2 D triangular PSP	8 benchmark sequence data set
PR	Custodio et al. [93]	Multiple minima GA, GA for PSP-HP problem: Chromosome: Absolute	PSP	Monomer sequence

		encoding. Population: Randomly generated; 500 chromosomes. Selection: Tournament selection of four randomly chosen individuals. Crossover: Standard two-point crossover, multi-point crossover. Mutation: Local move and loop move. Exhaustive search mutation and segment mutation. Replacement: Parental replacement with crowding. GA for atomistic PSP problem Chromosome: Structures with the same length of target sequences. Population: Not completely random; 500 chromosomes. Crossover: Standard two-point crossover, multi-point crossover. Mutation: Incremental mutation, compensatory mutation and segment mutation, TC operator. Replacement: Parental replacement with crowding.		data set

Notes: ODB, OMICS data bank; MOGAMOD, Multi-objective genetic algorithm for motif discovery; NSGA, non-dominated sorting genetic algorithm; PBIL, population based incremental learning; ACO, ant colony optimization; MRMR, minimum redundancy maximum relevance.

1.3.3 Data mining

DM is a process of analysis, interpretation and mining of information that could help in decision making. It includes different types of clustering methods, DT, KNN, association rules, a-priori algorithm and many others. The DM methods are employed for characterization, pattern matching, meta rule guided mining, clustering, data visualization, generalization, evolution, association, and classification. The summary of DM based bioinformatics systems is shown in table 1.4.

Table 1.4: DM based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
GE	Beeren-winkel et al. [53]	**Classification**: C4.5 and SVM. C4.5, a heuristic divide and conquer strategy, reduced error pruning. SVM, linear decision unction, Langrangian dual, Joachim's SVM. **Performance evaluation**: LOOCV technique.	Interpreting genotypic HIV drug resistance tests	Clinical samples data set and first 220 to 250 amino acids
TR	Xu and Zhang [76]	**Gene selection**: Virtual gene, linear combination of genes. **Classification**: KNN, DLD, SVM.	Gene selection and cancer classification	Colon, leukemia and multi-class cancer data sets
TR	Jiang and Gruen-wald [94]	Association rule: FIS-Tree mining. Use of FIS-Tree and BSC-Tree as data structures. FIS-Tree: Value is symbolized as an exponent bit, fraction bit and a sign bit, use of logical AND operations. BSC-Tree: Used for real time	Mining of microarray gene expression data	Data sets from Stanford University and Harvard Medical School

		compression for a bit string, “data mining ready” data structure, bottom-up model.		
TR	Zhang and Deng [55]	**Gene preselection**: Family wise error rate. **Gene selection**: Bhattacharyya distance, sequential forward selection algorithm. **Classification**: Linear SVM, KNN.	Gene selection and cancer classification	Colon, DLBCL, leukemia, prostate and lymphoma cancer data sets
TR	Priscilla and Swamynathan [95]	2-D hierarchical clustering, multilevel microarray clustering, semi-supervised, self clustering of each gene type in a vertical direction and bottom up hierarchical clustering in horizontal direction.	Gene expression data clustering	Leukemia, Adenocarcinoma and Lymphoma data sets
PR	Chmielnicki and Stapor [96]	**Classification**: SVM, RDA and SVM-RDA. SVM: LIBSVM, RBF kernel, OVO strategy with min- max voting scheme. **Feature selection**: Brute force algorithm, sequential forward selection, modified sequential forward selection. **Model validation**: Paired t-test, k-fold CV technique.	Protein fold recognition and structure prediction	Structural Classification of Proteins
PR	Abbasi et al. [81]	**Classification**: KNN, MLP, RBF and FRAN.	Protein fold recognition	PDB-40D data set

Notes: SVM, support vector machine; LOOCV, leave-one-out cross-validation; DLD, diagonal linear discriminant; FIS-Tree, frequent itemset tree; BSC-Tree, bit string compression tree; KNN, k nearest neighbor; DLBCL; diffuse large B-cell lymphoma; LIBSVM, library of support vector machine; RBF, radial basis function; OVO, one-versus-all, FRAN, fuzzy resource allocating network.

Different types of DM methods provide assorted advantages. DTs are efficient in handling discrete data [53]. KNN provides low comprehensibility. The integration of KNN with statistical technique provides better classification performance and evades the singularity problem linked with the within-class scatter matrix [47]. Association rule mining tree such as Frequent Itemset Tree (FIS-Tree), apriori algorithm and FP-Growth are used to find correlations among items in a given data set. FIS-Tree mining shows better performance than Apriori algorithm and FP-Growth [94]. Apriori algorithm saves search space and execution time [94]. FIS- Tree mining is advantageous as it shows generality and performs 800 times quicker than Apriori algorithm and 2 times quicker than FP-growth [94].

1.3.4 Fuzzy logic

FL uses the phenomenon of ‘more or less’ rather than ‘either–or’. FL accepts noisy and imprecise input and constitutes three parts: fuzzification, fuzzy inference and defuzzification. In fuzzification, a linguistic/fuzzy variable defines the concept of fuzzy logic. A process that scales and maps the real input variables to fuzzy sets is called fuzzification. The fuzzy inference engine consists of fuzzy if–then rules. It contains aggregation and composition operators. The former involves the computation of ‘IF’ part and the latter involves the computation of ‘THEN’ part. Finally, the defuzzification involves the conversion of fuzzy output values to control signals. The summary of the bioinformatics applications of FL is shown in table 1.5.

FL can cope very well with the noisy, inexact and missing data [98]. It offers user-friendly predictions and classification which are easily understandable by humans. It is a highly efficient and effective technique while dealing with uncertainty and vagueness of expression levels. Besides these advantages, FL is less utilized by the researchers because it offers poor knowledge elicitation and higher annotation ratio. It is very week in processing microarray data sets and quantitative indices [99].

Table 1.5: FL based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
GE	Ma and Chen [97]	Quantitative gene expression data on linguistic variable, use of fuzzy sets, membership functions, *n*th order fuzzy dependency relationship, standardized residual.	Discovery of gene regulatory networks for time series gene	S. Cerevisiae genes, Alpha, CDC15 and CDC28 data sets

			expression data	
GE	Ma and Chen [98]	Incremental fuzzy mining, Fuzzification: Fuzzy linguistic variables, 3 fuzzy sets. Fuzzy association pattern discovery: Use of standardized residual, adjusted residual. Weight assessment: Weight of evidence measure, probabilistic measure. Gene function prediction: Searching of fuzzy association patterns, weight of evidence supporting the assignment, merge all the indications granted by fuzzy association patterns, the calculation of degree of membership. **Performance evaluation**: 10-fold CV technique.	Gene function prediction	Yeast Genome and Munich Information Center for Protein sequences functional catalogue database of 52 MIPS functional classes, a data set of 517 genes
TR	Maji and Paul [99]	Cluster is symbolized by centroid, a possibilistic lower approximation and a probabilistic boundary, alternating optimization of an objective function.	Gene expression data clustering	14 yeast microarray data set
PR	Abbasi et al. [81]	**Classification**: KNN, MLP, RBF and FRAN.	Protein fold recognition	PDB-40D data set

Notes: CV, cross validation; KNN, k nearest neighbor; MLP, multi layered perceptron; RBF, radial basis function; FRAN, fuzzy resource allocating network.

1.3.5 Swarm intelligence

SI is the discipline that handles systems made up of a set of distributed and self-organized individuals. It is usually dependent up on the natural phenomenon [100]. The goals of SI techniques are self-organizing, robustness and performance optimization. SI includes artificial bee colony (ABC), particle swarm optimization (PSO), artificial immune system

(AIS), ant colony optimization (ACO), gravitational search algorithm (GSA), and many others. The summary of utilization of SI techniques in bioinformatics is shown in table 1.6.

Table 1.6**:** SI based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
GE	Gonzalez-Alvarez et al. [100]	Individual: Initial location of the potential motif on all the sequences, motif length. MO ABC, Population: 200 individuals, ranking using non-dominated sort and crowding distance. Selection: Greedy selection with the dominance concept. Mutation: Probability 0.8. Scout bees: 1. MO GSA, use of Newtonian Physics Theory. Population: 200 individuals, ranking using non- dominated sorting and linear bias. GO: 100.	DNA motif discovery, i.e., motif length, support and similarity	TRANSFAC
GE	Santander -Jimenez and Vega-Rodriguez [101]	MOABC, Individual: Phylogenetic topology, branch length value and parameters of evolutionary model. Initialization: Combination of employed bees and onlooker bees. Swarm size: 100.	Phylogenetic Inference	8 real nucleotides data sets and a real data set of salamander mitochond-rial DNA

		Maximum generations: 100. Mutation: 5%. Limit: 15.		
TR	Mohamad et al. [102]	EPSO, Particle: A binary string of length of total no of genes. Use of scalar quantity, i.e., particle speed and modified sigmoid function. **Performance evaluation**: LOOCV technique.	Gene selection	Leukemia and mixed lineage leukemia cancer data sets
TR	Wei et al. [103]	BPSO, Particle: Has position and velocity, shifts in a state space limited to 0 or 1 in each bit. Initialization: Random. Population: 50 particles. Termination: 2 0 0 iterations.	Identification of SNPs associated with Graves' disease	Human DNA sequence data set containing the genes CD28, CTLA4 and ICOS
PR	Nemati et al. [91]	**Feature selection**: GA and ACO runs in parallel. GA, Population: 50 chromosomes. Crossover: Probability, 0.7. Mutation: Probability, 0.005. Iterations: 100. ACO, Population: 50 chromosomes. Initial Pheromone: 1.	Feature selection in protein function prediction	GPCR-PROSITE and ENZYME-PROSITE data sets from UniProt and Prosite database

		Importance of pheromone level: 1. Importance of heuristic information: 0.1.		
PR	Li et al. [104]	IF-ABC, Selection: Replacement of roulette selection strategy by the parameter trial. Crossover and mutation: Variable number of coordinates and multi-point crossover. Exploitation: Enhanced by introducing a convergence factor in the crossover process.	Protein secondary structure optimization	Artificial fibonacci sequences and natural sequences from PDB

Notes: MO, Multi-objective; ABC, artificial bee colony; GSA, gravitational search algorithm; TRANSFAC, transcription factor database; EPSO, enhancement of binary particle swarm optimization; BPSO, binary particle swarm optimization; TLC, two-layer linear classifier; IF-ABC, internal feedback based on artificial bee colony.

ACO is an efficient, adaptive and robust search process method. It is easy to implement in less computation time and provides local searching, quick convergence and intelligent background [91]. Gravitational search algorithm (GSA) offers good scaling capability [100]. ABC works sound in exploration and privileged in exploitation [104]. PSO considers both global and local search capabilities and provides premature convergence [104]. AIS gives advantages when only the normal data is on hand [105].

1.4 Integrated Methods in Bioinformatics

The integration of various methods has been found in literature for solving various problems in hand. These integrated methods provide various advantages as compared to the individual methods. The integrated methods exploit the advantages and mitigate the disadvantages of each other. The applications of various integrations, i.e., ANN-SI, ANN-EA-RF, DM-LDA, ANN-PCA-PLS,ANN-DM-RF,ANN-DM-FL, ANN-EA-FL-SI, ANN-EA-FL, ANN-FL, CBR-DM, ANN-PCA, DM-SI, EA-FL, ANN-DM, ANN-EA, ANN-Bayesian classifier, DM-EA, EA-SI, ANN-CBR-DM, ANN-DM-EA, ANN-DM-PCA and ANN-EA-PCA are shown in tables 1.7 to 1.28 respectively.

Table 1.7**:** ANN-SI based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
PR	Abbasi et al. [81]	**Classification**: KNN, MLP, RBF and FRAN. MLP, 3 layered perceptron, nonlinear activation function and max operator. **Tuning of parameters of RBF**: PSO.	Protein fold recognition	PDB-40D data set

Notes: KNN, k nearest neighbor; MLP, multi layered perceptron; RBF, radial basis function; FRAN, fuzzy resource allocating network; PSO, particle swarm optimization, PDB, protein data bank.

Table 1.8**:** ANN-EA-RF based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
TR	Tong and Schierz et al. [42]	**Feature extraction**: GA, Chromosome: 10 genes; real number representation. Population: 300 chromosomes. Fitness computation: 3 layered FF MLP, 10-5-2, 10-5-3, 10-5-4 architecture. 67-79 nodes, where 7-9 nodes are bias nodes. Hyperbolic tangent activation function. Selection: Tournament selection, tournament size: 2. Crossover: Single point crossover, probability: 0.2.	Feature extraction for cancer classification	ALL/ AML and SRBCTs datasets

		Mutation: Probability: 0.1. Replacement: Elitism scheme. Termination: Evaluation size: 30000 and whole cycle repeat: 5000. **Classification**: MLP, SVM and RF. **Performance evaluation**: 10-fold CV technique.		

Notes: GA, genetic algorithm; FF, feed-forward; MLP, multi layered perceptron; SVM, support vector machine; RF, random forest; CV, cross validation.

Table 1.9**:** DM-LDA based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
TR	Sharma and Paliwal [47]	**Gene extraction**: GLDA operates in supervised mode, gradient descent algorithm, and an iterative algorithm. **Classification**: KNN.	Gene extraction and cancer classification	Acute leukemia, SRBCT and lung adenocarcinoma cancer data sets

Notes: GLDA, gradient linear discriminant analysis; KNN, k nearest neighbor.

Table 1.10: ANN-PCA-PLS based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
TR	Zeng et al. [59]	**Feature selection**: REDISC, supervised method. **Feature extraction**: PCA and PLS. **Classification**: Linear SVM, 2-norm	Gene selection, extraction and classification	Colon and Leukemia cancer data sets

		soft margin SVM. **Performance evaluation**: 10-fold CV technique.		

Notes: REDISC, redundancy elimination based on discriminative contribution; PCA, principal component analysis, PLS, partial least square, SVM, support vector machine, CV, cross validation.

Table 1.11**:** ANN-DM-RF based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
PR	Aram and Charkari [61]	Individual method: Two-layered classification framework. 1st layer: Classification employing RF, ordinary MLP and polynomial kernel SVM. 2nd layer: Instances into 27 folds. Fusion method: Two-layered classification framework and hierarchical learning architecture. **Classification**: KNN, MLP, RBFN, NB.	Protein fold recognition	Protein database

Notes: RF, random forest; MLP, multi layered perceptron; SVM, support vector machine; KNN, k nearest neighbor; RBFN, radial basis function network; NB, naïve bayes.

Table 1.12**:** ANN-DM-FL based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
TR	Wang et al. [66]	**Feature ranking**: T-test and class seperability. **Classification**: FNN and SVM. FNN, rule-base simplification, 4 layered architecture.	Cancer classification	Lymphoma, SRBCT, liver and GCM data sets

		Group of C-SVMs with RBF. **Filling of missing value**: K NN. **Performance evaluation**: 5-fold CV technique.		

Notes: FNN, fuzzy neural network; SVM, support vector machine; KNN, k nearest neighbor.

Table 1.13**:** ANN-EA-FL-SI based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
TR	Lee [67]	**Gene selection**: Regression analysis. **Selection of gene markers**: SVM, GA and PSO. SVM, RBF kernel function. GA, Chromosome: Human cDNA clones, binary encoding. Population: Randomly generated 20 chromosomes. Selection: Roulette wheel selection employing elitism strategy. Crossover: Two-point crossover, rate: 0.7. Mutation: Probability: 0.02. Termination: 100 generations.	Gene selection and cancer classification	Ovarian and breast cancer data sets

		PSO, Particle: Human cDNA clones. Population: Randomly generated 20 particles. Selection: Use of sigmoid function and inertia weights. Termination: 100 generations. **Extraction of gene markers**: ANOVA. **Classification**: Fuzzy if-then rules, Gaussian membership functions.		

Notes: SVM, support vector machine; GA, genetic algorithm; PSO, particle swarm optimization; cDNA, complementary DNA; ANOVA, analysis of variance.

Table 1.14: ANN-EA-FL based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
TR	Korfiati et al. [69]	**Filtering**: miRanda **Classification**: SVM and GA. SVM, LIBSVM library, RBF kernel. GA, Chromosome: Feature genes and parameter genes. Population: 50 chromosomes. Selection: Rank-based roulette wheel selection.	miRNA target prediction	miRBase, TarBase and miRecords datasets

		Crossover: 2-point crossover; probability: 0.9. Mutation: Dynamic control of mutation parameters. Maximum number of generations: 200. **Extraction of interpretable fuzzy rules**: Evolutionary support vector fuzzy inference system, if-then rules, RBF kernel, optimization of parameters using GA (500 generations and 30 population size). **Performance evaluation**: 5-fold CV technique.		

Notes: SVM, support vector machine; GA, genetic algorithm; LIBSVM, library of support vector machine; RBF, radial basis function.

Table 1.15**:** ANN-FL based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
TR	Azujae [74]	**Classification**: Supervised SFAM, 2 layered network architecture, **Learning process**: Fuzzy logic operations. **Performance evaluation**: LOOCV technique.	Prediction and discovery of classes of cancer	LLMPP data sets
TR	Anandakumar and Punithavalli [106]	**Gene importance ranking**: ANOVA.	Gene selection and cancer classification	Lymphoma and liver cancer data

		Minimum gene subset: Two fuzzy if-then rules, 5 layered architecture, Learning using modified Levenberg-Marquardt algorithm. **Classification**: Fast adaptive neuro-fuzzy inference system.		sets
Functional Genomics	Neague and Palade [107]	Fuzzy if-then rules. Multilayered neural structure, 3 layered FF network, MLP. Integration of FEMF, UGN, SGN. FEMF, UGN, SGN uses max fuzzy operator, softmax transformation, supervised training neural network, respectively.	Functional analysis of gene expression data	E. Coli data set

Notes: SFAM, simplified fuzzy art map; LOOCV, leave-one-out cross validation; ANOVA, analysis of variance; FF, feed-forward; MLP, multi-layered perceptron.

Table 1.16**:** CBR-DM based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
TR	Yao and Li [108]	**Gene preselection**: Nonparametric scoring algorithm. **Sample clustering**: Hierarchical clustering. **Gene selection**: ANMM, nearest between-class distance maximization and furthest within cluster distance minimization, nonparametric discriminant analysis. **Classification**: CBR,	Gene selection and classification	Simulated data set and real data set composed of leukemia, colon, SRBCT and GCM Cancer

		Rule: Define the domain knowledge. Retrieve: Small distant cases from m- Dimensional vector will be retrieved. Reuse: Minimum distance between m- Dimensional vector and case base.		

Notes: ANNM, Additive Nonparametric Margin Maximum; CBR, Case-Based Reasoning.

Table 1.17: ANN-PCA based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
GE	Nikolova et al. [109]	**Preprocessing**: PCA. **Prediction**: ANN, **ANN architecture**: 70-105-2 architecture, FF networks. **Learning**: Supervised learning approach. **Activation function**: Sigmoidal non linearity hyperbolic tangent activation function, Hidden layer: tan-sigmoid activation function Output layer: Pureline.	DNA sequence prediction	Protease gene of HIV-1 virus data set
TR	Peterson and Ringer [110]	**Preprocessing**: Apply cuts on intensities and spot areas. **Dimension reduction**: PCA. **Classification**: MLP; supervised	Tumor classification	SRBCT and breast cancer data sets

		learning, and 8-4-1 unit architecture. **Performance evaluation**: 3–fold CV technique.		
TR	Ao and Ng [111]	**Feature extraction**: PCA. **Prediction**: ANN, **Architecture**: 3 layered architecture, one hidden layer with 10, 5 and 20 hidden neurons and AIC method. **Activation function**: Tan-sigmoid activation function.	Modelling of gene expression time series	Yeast gene expression levels and other genes data set
TR	Ziaei [112]	**Gene ranking**: Signal to noise ratio. **Dimension reduction**: PCA. **Classification**: LP, 10-1unit architecture.	Cancer classification and prediction of a class of Lymphoma	40 patients and 4026 gene expression level dataset
TR	Chen et al. [113]	**Normalization and computation**: MAS5 function. **Preprocessing**: Correlation coefficient, RankProd and PCA. **Feature selection**: Chi-square test. **Prediction**: ANN, **Architecture**: Standard FF, fully connected, BP MLP, 3 layered architecture and supervised training. **Evaluation**: Kaplan-Meier survival	Cancer patient survival prediction	NSCLC data and NCI caArray

		analysis and log-rank test.		

Notes: PCA, principal component analysis, ANN, artificial neural network; FF, feed-forward; MLP, multi-layered perceptron; CV, cross validation; FF, feed-forward; BP, back propagation; MLP, multi-layered perceptron.

Table 1.18**:** DM-SI based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
TR	Chen et al. [114]	**Gene selection**: PSO, Particle: binary encoding. Initialization: Random. Fitness Function: Calculated using C4.5.	Gene selection and cancer identification	11 cancer data sets
TR	Kar et al. [115]	**Gene selection**: PSO, Initial positions: Random; Fitness function: Calculated using KNN. **Classification**: SVM and K NN with K=3 to 20 with the help of a heuristic approach. **Performance evaluation**: 3-fold CV technique.	Gene selection and cancer classification	SRBCT, ALL-AML and MLL cancer data sets
PR	Turkoglu and Kaymaz [116]	**Reduction of data dimension**: AIS, aiNET algorithm in supervised manner, euclidean distance, affinity maturation process. **Classification**: KNN with K=7, lazy learning process.	Prediction of protein cellular localization sites	E. Coli datasets

Notes: PSO, particle swarm optimization; KNN, k nearest neighbor; SVM, support vector machine; CV, cross validation; AIS, artificial immune system.

Table 1.19**:** EA-FL based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
GE and TR	Jacob et al. [117]	FGA: GA-Fuzzy fitness finder. Fuzzy fitness finder, Fuzzification: Set of values. Fuzzy membership functions: Triangular sets. Fuzzy inference: Fuzzy if-then rules. Defuzzification: Root sum squares method. GA, Selection: Roulette wheel selection. Population: 10 chromosomes, not random, probability: 1. Mutation: Local search method, probability: 0.1. Iterations: 50.	Sequence segmentation and prediction of functionally related genes	E. Coli, bacillus subtilis and mycobacterium tuberculosis dataset
TR	Schaefer and Nakashima [118]	**Classification**: Fuzzy if-then rules, triangular fuzzy sets and linguistic rules. Hybrid fuzzy classification: Fuzzy if then rules-GA. Fuzzy if-then rules: A-priori defined number of rules which are randomly generated as an initial population.	Analysis of gene expression data	Colon, leukemia and lymphoma cancer data sets

		GA, Michigan style algorithm, Chromosome: S tring. Population: 20 chromosomes. Selection: Binary tournament selection with replacement. Crossover: Uniform crossover, probability: 0.9. Mutation: Probability: 0.1. Generation update: Elitist strategy. Generations: 30000. The selection probability of don't care attributes: 0.5.		
TR	Wang and Palade [119]	**Gene selection**: FCCEGS, weighted fuzzy if-then rules, 15 fuzzy sets. **Selection of small set of rules**: MOEA, Chromosome: Binary encoding. Population: Three objectives: weight vector, no of selected fuzzy rules, total no of antecedent conditions. Crossover: Simple 2-point crossover. Mutation: Biased mutation, probability from 1 to 0 and from 0 to 1.	Gene selection and cancer classification	Lung, ovarian and colon cancer data sets

TR	Zibakhsh and Abadeh [120]	**Pattern classification**: Fuzzy if-then rules coded as a string, set of linguistic values, use of 6 linguistic variables. **Classification**: Memetic algorithm, Chromosome: Fuzzy if-then rule. Population: Randomly created 50 fuzzy if-then rules. Fitness function: Global fitness function and local fitness function. Selection: Roulette wheel selection. Crossover: Uniform crossover; probability: 0.9. Mutation: Random mutation method; probability: 0.1. Replacement: 20%.	Gene selection and cancer/ tumor detection	Tumor datasets
PR	Mansoori et al. [121]	**Feature reduction**: Feature ranking algorithm. **Generation of rules**: SGERD, fuzzy if- then rules, triangular membership functions, use of product operator.	Protein sequence classification	UniProt

Notes: GA, genetic algorithm; FCCEGS, Fuzzy C-Mean Clustering based Enhanced Gene Selection method; MOEA, multi-objective evolutionary algorithm; SGERD, Steady-state genetic algorithm for extracting fuzzy classification rules from data.

Table 1.20: ANN-DM based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
GE	Kasabov and Pang [122]	**Search**: Motif search engine. **Judgment**: KNN. **Classification**: Ensemble of SVM based on majority voting and transductive SVM. **Performance evaluation**: 3-fold CV technique.	Promoter recognition	EPD and GenBank
TR	Simek et al. [123]	**Preselection**: Sebestyen criterion and correlation coefficient. **Gene selection**: SVD and RFR. **Clustering**: Hierarchical complete linkage clustering algorithm. **Classification**: Linear SVM. **Performance evaluation**: LOOCV technique.	Clustering, classification, feature selection and modelling of the dynamics of gene expression data	Tumor/ normal thyroid microarray dataset and Yeast CDC-15 data set
TR	Zheng et al. [124]	**Preselection**: T-statistics. **Clustering**: IVGA principle, heuristic combinatorial optimization method and variational Bayesian learning. **Classification**: SVM with RBF kernel. **Performance evaluation**:	Gene selection and tumor classification	Colon, acute leukemia and prostate cancer data sets

		LOOCV technique.		
TR	Bose et al. [125]	**Object selection**: MRMR. **Gene extraction**: Partition based attribute clustering algorithm, supervised way. **Classification**: NB, SVM and KNN.	Gene extraction and classification	Colon, lung and leukemia cancer data sets
PR	He et al. [126]	**Preprocessing and training**: SVM, 3 OVA binary classifiers. **Extraction of rules**: C4.5.	PSSP	RS126

Notes: KNN, k nearest neighbor; SVM, support vector machine; CV, cross validation; SVD, singular value decomposition; RFR, recursive feature replacement; LOOCV, leave-one-out cross-validation; IVGA, independent value group analysis; RBF, radial basis function; MRMR, minimum redundancy maximum relevance; NB, naïve bayes.

Table 1.21**:** ANN-EA based bioinformatics system

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
TR	Shanthi et al. [127]	**Feature selection**: GA, Chromosome: A sequence of consecutive genes. Population: 20 chromosomes. Selection: Roulette wheel selection. Crossover: Arithmetic crossover; probability: 0.6. Mutation: Non-uniform mutation; rate: 0.033. Number of generations: 20.	Feature selection for diagnosis of stroke diseases	150 patient data set

		Relationship between input and output: ANN, **Architecture**: 14-7-10 unit architecture, backward stepwise algorithm, BP algorithm, sigmoidal function, output activation function.		
TR	Chen and Hsu [128]	**Gene selection**: Signal to noise ratio and GAGS. Chromosome: Consists of 25, 50, 100 and 150 genes. Population: 40 chromosomes. Reproduction: Reproduction rate at 40% in primary stage and at 20% in last stage. Crossover: Single-point crossover, random generation, two chromosomes are randomly selected. Mutation: Optimal solution, high mutation possibility in primary stage and low mutation possibility in last stage. **MTSVSL** MTL: Inductive transfer mechanism, shared perception, BP network, learning in both main task and shared task. SVS: Hyperplane. **Performance evaluation**: Random average 3-fold CV	Gene selection and cancer classification	Leukemia and prostate cancer data sets

		technique.		
TR	Akadi et al. [129]	**Feature selection**: MRMR, preprocessing of high-dimensional microarray data. GA, Chromosome: Gene encoding. Population: 100 chromosomes; randomly chosen. Crossover: Probability 0.8. Mutation: Probability 0.1. Termination: 20 generations. **Classification**: SVM and NB. **Performance evaluation**: LOOCV technique.	Feature selection	NCI, lymphoma, lung, leukemia and colon cancer data sets
PR	Otwani et al. [130]	**Optimization of topology of ANN**: GA. **Prediction**: Fully connected FFBPN.	PSSP	PDB
PR	Li et al. [131]	**Feature selection**: Statistical testing, GA. GALOPPS: Chromosome: Vector of integers with range 0-9999. Population: 200 chromosomes. Selection: Stochastic universal sampling. Crossover: 2-point crossover, probability 0.5.	Gene selection and cancer classification	3 Serum SELDI MS data sets

		Mutation: Multi-field mutation, probability 0.02. Generations: 8000. Termination: Till the maximum no of generations or value of fitness function has reached 1. **Classification**: SVM, polynomial kernel function with d=1. **Performance evaluation**: LOOCV technique.		
PR	Reyaz-Ahmed [132]	**Encoding**: Orthogonal encoding with BLOSUM62 matrix. SVM, OVA and OVO binary classifiers, RBF kernel. GA, Chromosome: binary encoding. ANN, **Architecture:** 4 layered architecture.	PSSP	RS126
PR	Nanni and Lumini [133]	**Generation of reduced alphabets**: GA, Chromosome: A string of amino acid of length 20. Population: Initial population is randomly generated; Population size=10. Selection: Cross-generational strategy.	Protein and peptide classification	HIV, peptide and vaccine datasets

		Crossover: Uniform crossover; probability 0.96. Mutation: Probability 0.02. Number of generations: 5. **Classification**: Linear SVM, and RBF SVM.		
PR	Li et al. [134]	**Preselection of features**: MRMR. **Optimization of feature sets**: GA, Chromosome: Binary and decimal coded genes. Population: 30 chromosomes. Selection: Elitist strategy. Crossover: Random positions. Mutation: Part of decimal coding. Termination: 10000 generation. **Classification**: SVM with RBF kernel, OVO strategy. **Performance evaluation**: 10-fold CV technique.	Prediction and classification of G-protein coupled receptors	Protein sequence data set
System Biology	Ritchie et al. [135]	Optimization of ANN architecture: Use of the binary expression tree, GP operators, set of rules. Chromosome: Binary expression tree encoded representation. Population: Initial random set of chromosomes.	Detection of nonlinear interactions among genes in common human diseases	Simulated data set

		Selection: Fitness proportionate selection and Roulette wheel selection techniques. Crossover: As per the rule of network construction. Termination: Till the maximum number of generations or classification accuracy 100%.		
System Biology	Noman et al. [136]	**Capture the interaction among genes**: Decoupled version of RNN, canonical RNN with delayed feedbacks, tightly coupled system, sigmoid function. **Identification of interactions**: DE, reverse engineering algorithm, Chromosome: Parameters for genes. Initialization: Random. Mutation: Random individuals selected. Crossover: Randomly inherited. Replacement: If find with better or the same fitness value. DE uses random restart strategy.	Reconstruction of gene regulatory network from gene expression	SOS DNA repair network of E. Coli

Notes: GA, genetic algorithm, ANN, artificial neural network, BP, back propagation; GAGS, genetic algorithm gene selection; MTSVSL, multitask support vector sample learning; MTL, multitask learning, CV, cross validation; MRMR, minimum redundancy maximum relevance; SVM, support vector machine; NB, naïve bayes; LOOCV, leave-one-out cross-validation; FFBPN, feed-forward back propagation network; RBF, radial basis function; OVO, one-versus-one; GP, genetic programming; RNN, recurrent neural network; DE, differential equation.

Table 1.22**:** ANN-Bayesian classifier based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
TR	Maglogiannis et al. [137]	Diagnosis using the probabilistic neural network. Prognosis using the generalized regression radial basis neural network **Classification**: SVM, Wolfe-Dual form, Gaussian RBF and polynomial kernel, OVA strategy, bayesian classifiers. **Performance evaluation**: 10-fold CV technique.	Prognosis and diagnosis of breast cancer	WDBC and WPBC data sets

Notes: SVM, support vector machine, RBF, radial basis function; CV, cross validation.

Table 1.23**:** DM-EA based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
TR	Lu et al. [138]	IGKA: Chromosome: Cluster number. Population: A set of Z coded solutions; 50. Selection: Proportional selection. Mutation: Mutates each allele to a new value; 0<MP<1. To speed up the convergence process: Classical k-means algorithm.	Gene expression data clustering	fig2data and cho data sets

		No of generation: 100.		
TR	Shah and Kusiak [139]	**Feature extraction**: WDTGS, GAGS, and feature set intersection approach. WDTGS: Production of rules, multiple users defined weighting schemes, 32 runs of DT. GAGS: GA-CFS, GA-DTW. GA-CFS: Chromosome: n genes. Population: 100 individuals. Selection: Threshold frequency of 60% was set for selection. Crossover: Crossover rate: 0.6. Mutation: Rate: 0.033. GA runs: 100. GA-DTW: Chromosome: n genes. Population: 100 individuals. GA runs: 100. Decision Tree: Building of 50000 decision trees. Feature sets intersection approach: Combination of feature set. **Performance evaluation**: 10-fold CV technique.	Gene/ SNP selection	Drug and Placebo data sets
TR	Han and Rao [140]	**Mining co-regulated clusters**: Association Rules, Item: Represented by 1, 0 or -1.	Mining co-regulated genes	SMD

		Reduction of running time: Hash-Tree, itemset in a unit hash tree. Node: Information includes a gene name, hash-table, support and pointers to the child node and increasing/ decreasing tendency. **Generation of rules**: GA, Chromosome: k-length sequence composed of 0 and 1. Population: 12 chromosomes. Selection: According to their arrangement of fitness. Crossover: Selected 2 positions randomly to crossover. Mutation: Selected random positions to mutate.		
TR	Wu [141]	**Clustering**: Genetic weighted k-means algorithm. Chromosome: A partitional string encoded by centers of clusters. Population: A set of partitional string; population size=21. Selection: Selection of individuals from the previous population. Crossover: Single point crossover. Mutation: Uniform replacement; probability 0.10.	Gene expression data clustering	Synthetic dataset and two real life gene expression data sets

		WKM Operator: k- means operators are employed.		
TR	Aljahdali and El-telbany [142]	**Informative gene set searching**: GA, steady state model. Chromosome: A set of 20 genes indices. Population: 50 random set of genes. Crossover: Probability 0.8. Mutation: Probability: 0.01. Population: Replacement of 25%. **Classification**: C 4.5.	Gene selection and cancer classification	NC160

Notes: IGKA, incremental genetic k-means algorithm; CV, cross-validation; GA, genetic algorithm.

Table 1.24**:** EA-SI based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
GE	Jangam and Chakraborti [143]	**Obtain a set of alignments**: ACO, Individual: Sequence Pair. Population: Even number of sequence pairs. Selection: Roulette wheel selection strategy. **Generation of accurate alignments**: GA, Individual: Alignment of nucleotide pair. Population: Alignments obtained	Alignment of two nucleic acid sequences	Nucleic acid sequence data set

		from ACO. Selection: Roulette wheel selection. Crossover: Multi-point crossover. Mutation: Random insertion or deletion of gaps. Termination: Repeated till convergence.		

Notes: ACO, artificial colony optimization; GA, genetic algorithm.

Table 1.25**:** ANN-CBR-DM based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
TR	Paz et al. [144]	Case: Information about the patient, rules, classification and probes marked as irrelevant or important. Retrieve: Preprocessing is done using RMA. Reuse: ESOINN, Competitive Hebbian Learning. PAM algorithm runs in parallel of ESOINN. Revise: Knowledge extraction using J48 algorithm. Retain: Correct and relevant decision rules generated.	Classification of microarray data	CLL patient data set

Notes: ESOINN, enhanced self-organized incremental neural network.

Table 1.26**:** ANN-DM-EA based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
TR	Anekboon et al. [145]	**Feature preselection**: Supervised neural network, MLP. **Best feature selection**: GA, Chromosome: A set of allele; variable length. Selection: A stochastic universal sampling with elitism strategy. Population: 2 sets. Crossover: Variable length crossover operation, rate: 0.5. Mutation: For short range chromosome, rate: 0.3. Termination: Number of iterations. **Classification**: MLP, SVM and DT. **Performance evaluation**: LOOCV and 5-fold CV technique.	Extracting predictive SNPs in Crohn's disease	Crohn's disease dataset
PR	Doong and Yeh [146]	**Data Encoding**: DSSP. **Clustering**: Partitional methods, GA. **Performance measure**: k-means. **Aligning the sequences**:	PSSP	CB513 and PDB

		Dynamic programming. **Classification**: RBF kernel and soft margin SVM.		
PR	Nanni and Lumini [147]	**Feature selection**: Sequential forward floating selection. **Clustering**: k-means. GA, Chromosome: A string of amino acid of length 20. Population: Randomly generated set of chromosome. Selection: Cross generational. Crossover: Uniform crossover, probability: 0.96. Mutation: Probability: 0.02. **Classification**: SVM with RBF kernel.	Prediction of bacterial virulent proteins	SWISS-PROT and VFDB datasets

Notes: MLP, multilayer perceptron; SVM, support vector machine; GA, genetic algorithm; DT, decision tree; LOOCV, leave-one-out cross-validation; CV, cross validation; RBF, radial basis function; SVM, support vector machine; RBF, radial basis function.

Table 1.27**:** ANN-DM-PCA based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
TR	Liu et al. [148]	**Resampling**: Bootstrap mechanism. **Feature extraction and selection**: Mann Whitney test, PCA, masked out clustering and	Feature, extraction, selection and cancer classification	Leukemia, lung, prostate, DLBCL, ovarian,

		t-test. **Classification**: 3 NN, 1 hidden layer FF networks, 10 hidden units and 1 output unit, soft voting mechanism. **Performance evaluation**: LOOCV and 10-fold CV technique.		colon and leukemia cancer data set
TR	Naplitano et al. [149]	**Preprocessing**: Noisy data rejection. **Feature extraction**: PCANN Based on STIMA algorithm, 1 layered FF, Hebbian type learning rules. **Clustering**: PPS, nonlinear and parametric mapping, hierarchical agglomerative clustering algorithm, uses Fisher's linear discriminant and Negetropy information.	Feature extraction and clustering	Human Cancer Cell Line

Notes: PCA, principal component analysis, NN, neural network; FF, feed-forward; LOOCV, leave-one-out cross-validation; CV, cross validation.

Table 1.28**:** ANN-EA-PCA based bioinformatics systems

Application area	**Author(s)**	**Intelligent method**	**Machine learning task**	**Database/ Databank**
TR	Karimi and Farrokhnia [150]	**Variable clustering**: SOM. **Dimension reduction**: PCA. **Variable selection and classification**: GA-LDA.	Selection, clustering and classification	Leukemia and SRBCT data set

		GA, Population size: 150. Selection: Roulette wheel selection. Crossover: 50%. Mutation: 1%. Number of generations: 200. **Performance evaluation**: LMOCV technique.		

Notes: SOM, self organizing map; PCA, principal component analysis; GA, genetic algorithm; LDA, linear discriminant analysis, LMOCV, leave-many-out cross-validation.

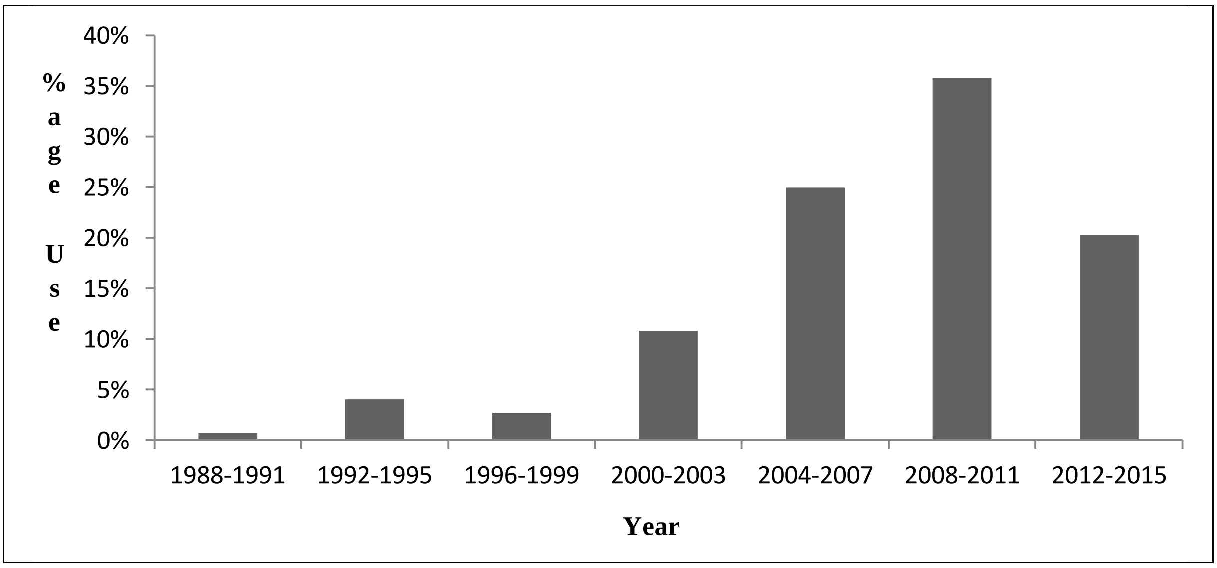
The integrated method ANN-DM works faster than individual linear SVM and polynomial SVM [51]. The integrated method ANN-EA-FL-SI provides superior classification performance and higher accuracy [67]. This integrated method is more statistically significant and efficient [42] [74]. The integrated method ANN-FL provides effective and efficient prediction results [74]. The integrated CBR-DM method provides robustness and performs better than SVM and KNN, when there is condition of highly noisy data [108]. The integrated method ANN-DM-SI is a simplified method for the easy diagnosis of cancer [115]. The integrated method DM-SI offers low complexity but it consumes more time for large datasets and the number of memory cells affects the classification performance [116]. The integrated method EA-FL provides higher interpretability, high search ability, less complexity and simple, smaller and more understandable rules, but it increases the computational overhead and their classification accuracy is lower than the alignment based approaches [118].

The integrated method ANN-EA increases the classification accuracy and improves the success rate [127]. The integration converges to global optimum faster than other methods [138] and sensitive to initial partitions and leads to high computational overhead [139] [141]. The integrated DM-EA method superior than k-means for the cluster quality and cluster sensitivity to primary partitions [141]. It can effectively create comprehensive trees and have high predictive power [142]. The integrated method ANN-CBR-DM facilitates detection, classification and reliable diagnosis [144]. The integrated method EA-SI possesses strong searching capability, ability to converge quickly and performs better than BLAST and ClustalW while aligning short and medium sized sequences. But it leads to higher time complexity when it deals with large sequences [91] [143]. The integrated method ANN-DM-EA tries to fill the performance gap between the amino acid sequences based feature and the evolutionary information based feature [147]. The integrated method ANN-DM-FL can find minimum gene subsets. This integration is simple, effective and offers high predictive accuracy [151]. The summary of the utilization of the integrated techniques is shown above.

1.5 Results and Discussion

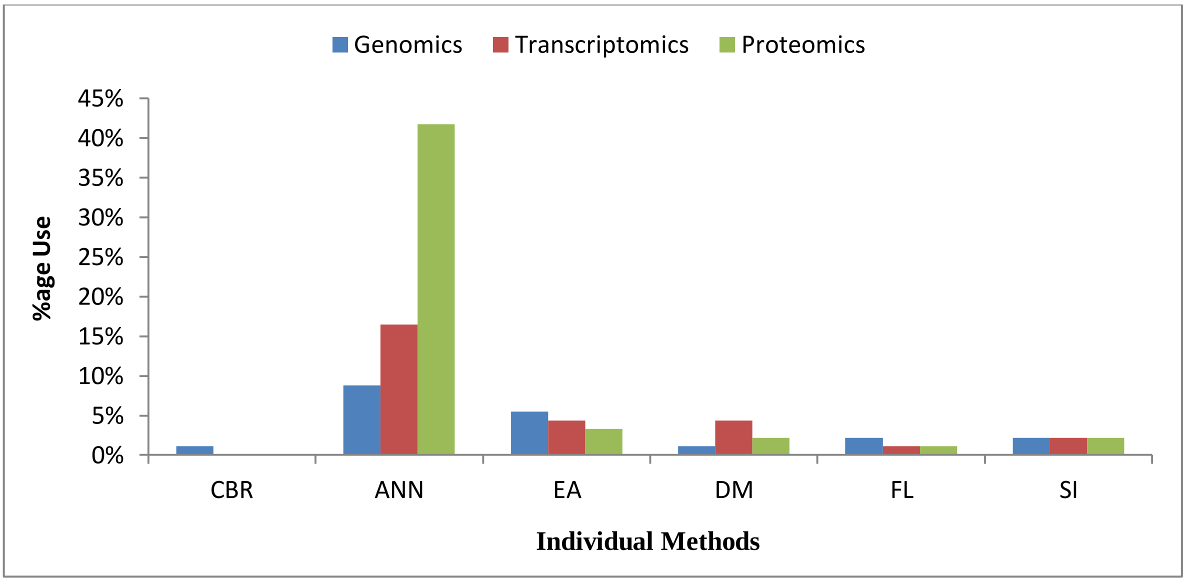
The results are shown year wise, deployment of individual and integrated methods in the areas of GE, TR and PR, utilization of methods for solving challenges in GE, TR and PR one by one. In figure 1.1, the %age use of all the individual and integrated methods for all the biological problems considered in the areas, i.e., GE, TR and PR are demonstrated from year 1988-2015. From the figure it is observed that the use of KBMs and ICMs is 4% in 1992-1995. The year range 1988-1991 shows the minimum use, i.e., 1%. The maximum use (36%) of KBS and ICT in bioinformatics is in the year 2008-2011 and the minimum use (approximately 4%) is in the year 1988-1999 whereas in moderation it is used (approximately 11%) in the year 2000-2003.

Figure 1.1: Year wise use



In figure 1.2, the %age deployment of individual KBMs and CIMs in GE, PR and TR areas are shown. It is observed that ANN is the most widely used methods in these areas because of its ease of implementation and less computational overhead. The application of ANN in GE is 9%, in TR is 16% and in PR is 42%, whereas the usability of EA in GE is 5%, TR is 4% and PR is 3%. DM is equally employed as an EA in case of TR, i.e., 4%, but less in GE and PR, i.e., 1% and2%. But very few literatures were found on employment of CBR, FL and SI.

Figure 1.2: Comparative views of utilization of individual KBMs and CIMs on Genomics, Transcriptomics and Proteomics



In figure 1.3, the % age uses of integration of various KBMs and CIMs for solving GE, TR and PR areas problems are illustrated. From the figure, it can be observed that the maximum number of applications deploys the integration EA-ANN. It was found to be 5% in TR and 9% in PR because the integration of these two methods improves the classification accuracy and provides robustness. The integrated methods like ANN-PCA, EA-FL, ANN-DM and EA-SI contributed a little part in the area of GE, i.e., 2%, whereas the integrated methods, namely ANN-SI, ANN-EA-RF, DM-LDA, ANN-PCA-PLS, ANN-DM-RF, ANN-DM-FL, ANN-EA-FL-SI, ANN-EA-FL, ANN-FL, CBR-DM, DM-SI, ANN-EA, ANN-Bayesian classifier, DM-EA, ANN-CBR-DM, ANN-DM-EA, ANN-DM-PCA and ANN-EA-PCA does not contributed anything in solving challenges of area GE.

In the area of TR, the integrated methods, namely ANN-EA-RF, DM-LDA, ANN-PCA-PLS, ANN-DM-FL, ANN-EA-FL-SI, ANN-EA-FL, CBR-DM, ANN-Bayesian classifier, ANN-CBR-DM, ANN-DM-EA and ANN-EA-PCA contributed just 2% in solving the challenges, and methods like ANN-FL, DM-SI and ANN-DM-PCA contributed 4% whereas the methods like ANN-PCA and EA-FL, both shows 8% and the best contribution is given by the integrated method DM-EA, i.e., 10%. The integrated methods ANN-SI, ANN-DM-RF and EA-SI were not implemented in area of TR. In case of PR, the best results are shown by ANN-EA and the methods like ANN-SI, ANN-DM-RF, DM-SI, EA-FL, ANN-DM, ANN-DM-EA also shows some contribution. Rest other methods does not show anything in the area of PR.

Figure 1.3**:** Comparative views of utilization of integrated KBMs and CIMs on Genomics, and Transcriptomics and Proteomics

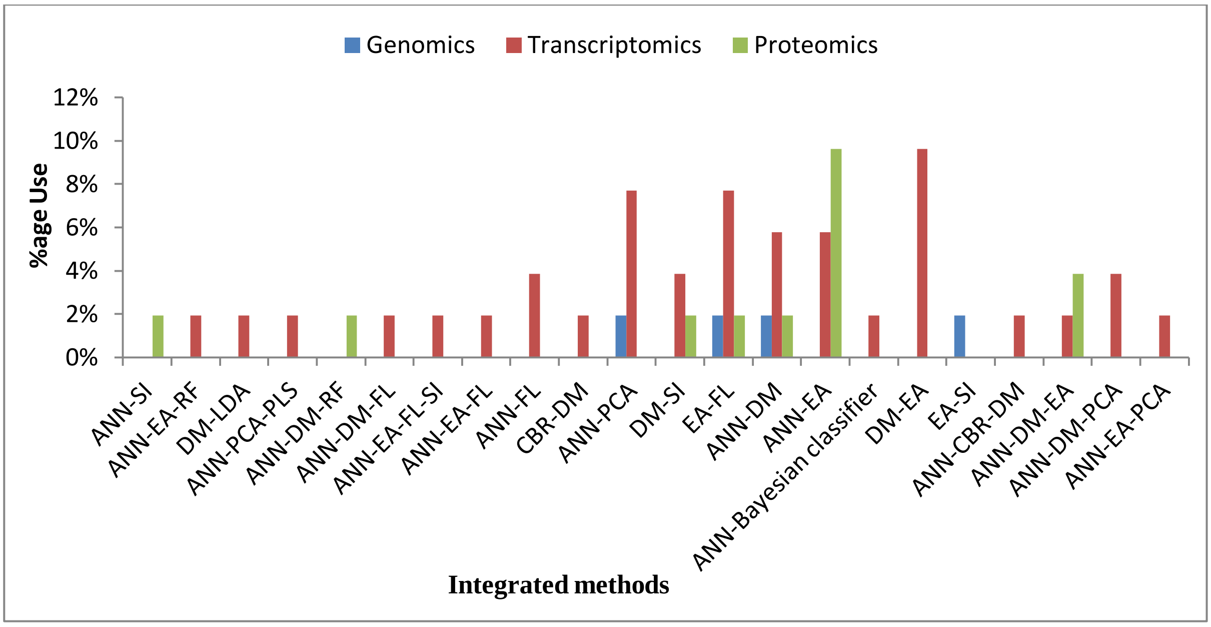
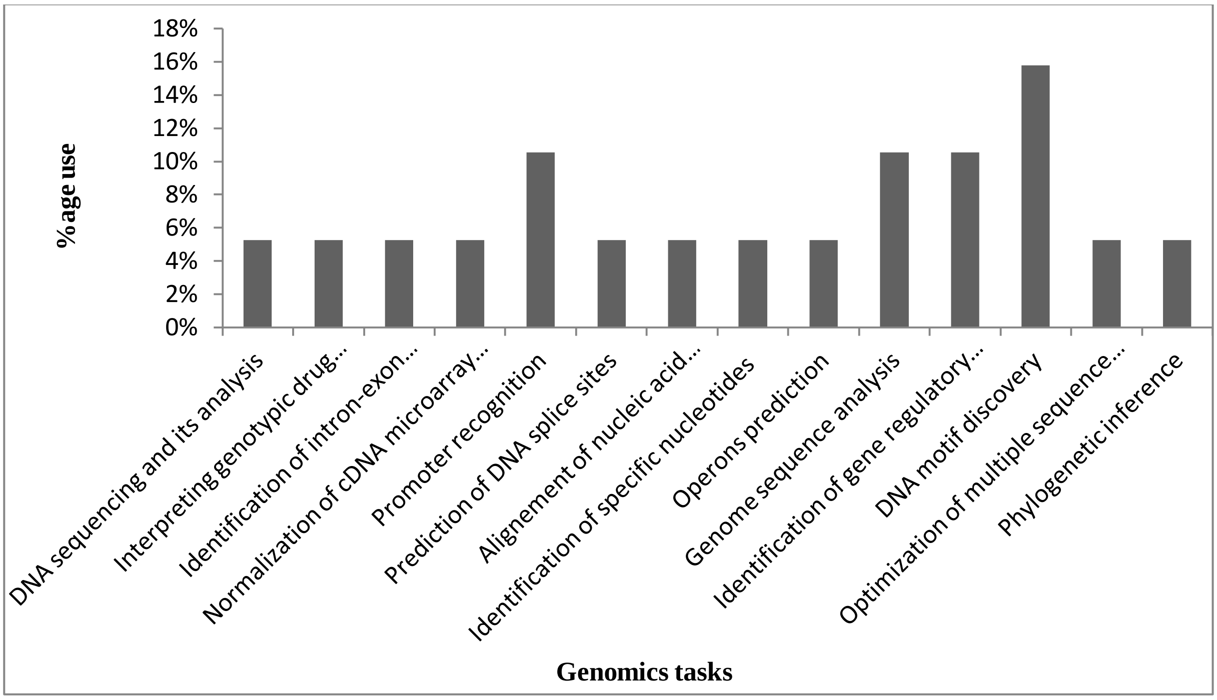


Figure 1.4 shows the %age use for solving the challenges in the area of GE. The individual and integrated methods are highly deployed for the discovery of DNA motifs, i.e., 16% of the all GE problems. For other problems like promoter recognition, genome sequence analysis and identification of gene regulatory networks, the methods employed are 11%. The review of literature shows that the complex tasks like DNA sequencing and its analysis, interpreting genotypic drug resistance tests, identification of intron-exon boundaries, normalization of cDNA microarray data, prediction of DNA splice sites, alignment of nucleic acid sequences, identification of specific nucleotides, operons prediction, optimization of

multiple sequence alignment and phylogenetic inference are also solved equally using these methods, i.e., 5%. In case of TR, the intricate task of gene selection and classification is solved using these methods and it shows a higher percentage of 47%. The other similar problems like gene extraction and classification; and gene expression data clustering and classification are solved 12% and 14% respectively. Sometimes, the task, i.e., dimension reduction and classification is solved in three phases for complex databases, it shows 5 %.

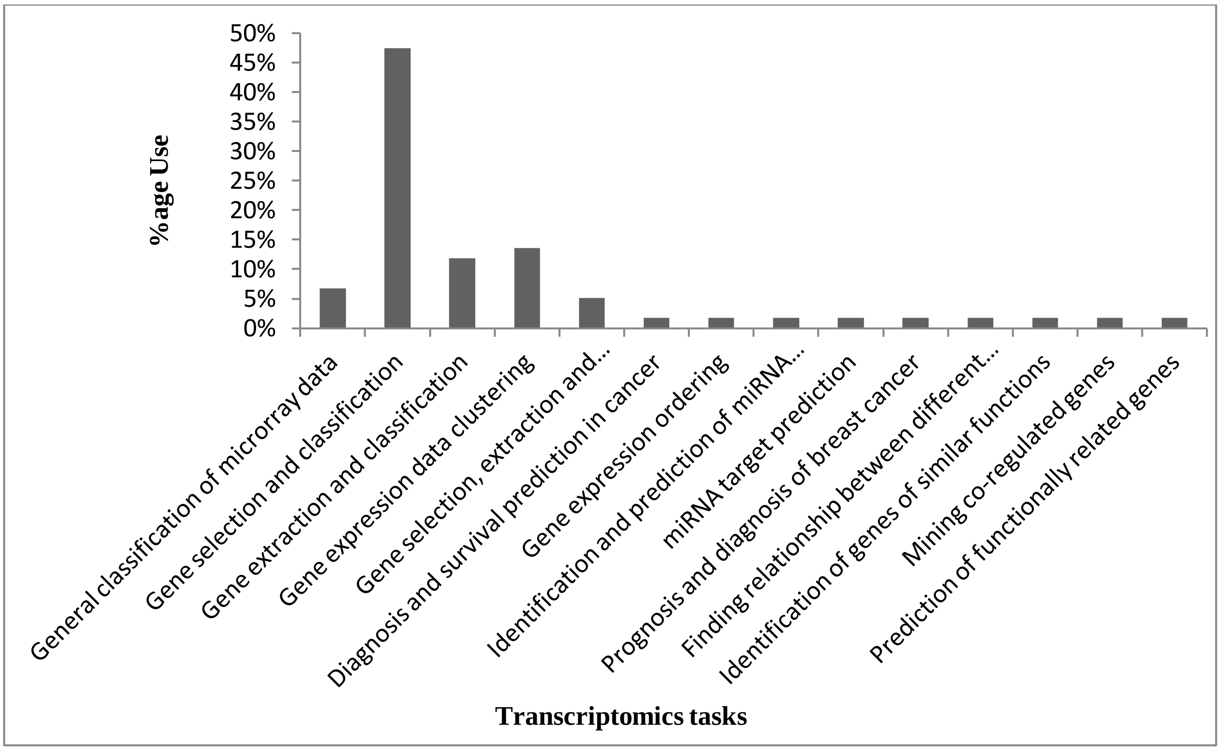
Figure 1.4**:** Comparative views of utilization of KBMs and CIMs for biological tasks in Genomics



The other challenges, namely diagnosis and survival prediction in cancer, gene expression ordering, identification and prediction of miRNA in viruses, miRNA target prediction, prognosis of breast cancer, finding relationship between different genes, identification of genes of similar function, mining co-regulated genes and prediction of functionally related genes shows 2%. The graphical view of each problem of TR is shown in figure 1.5.

Figure 1.6 shows the %age usability of these techniques for solving the problems in the area of PR. The most typical problem in PR, i.e., protein structure prediction is solved using the KBMs and CIMs and it shows 60%.

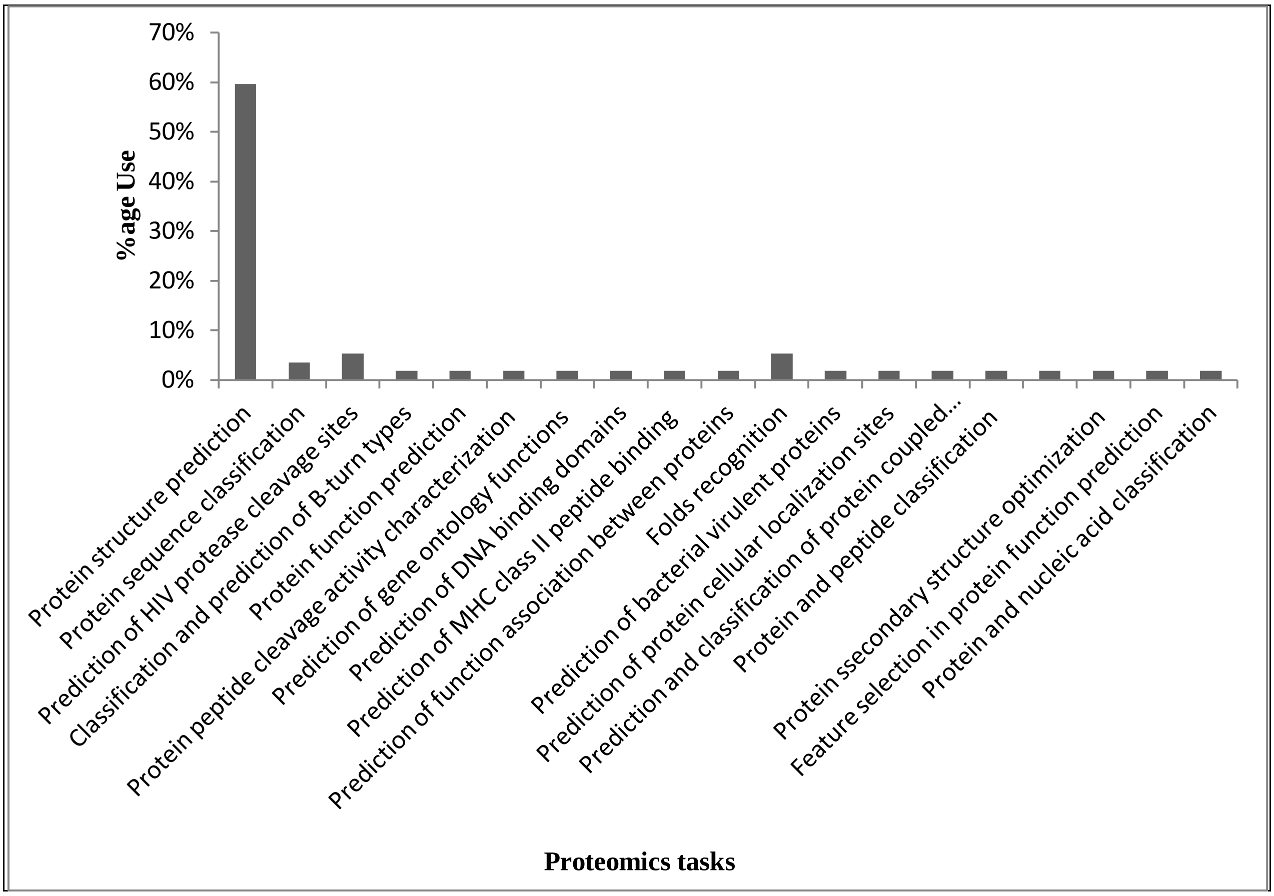
Figure 1.5**:** Comparative views of utilization of KBMs and CIMs for biological tasks in Transcriptomics



Three challenges, namely protein fold recognition, prediction of protease cleavage sites and protein sequence classification illustrates 5%, 5% and 4% respectively. All other challenges such as classification and prediction of B-turn types, protein function prediction, protein peptide cleavage activity characterization, prediction of gene ontology functions of proteins, prediction of DNA binding domains in proteins, prediction of MHC class II peptide binding, prediction of functional association between proteins, prediction of bacterial virulent proteins, prediction of protein cellular localization sites, prediction of protein coupled receptor, protein and peptide classification, diagnosis of disease using serum proteomic profiling, protein secondary structure optimization, feature selection in protein function prediction and protein and nucleic acid classification just shows 2%.

From the above results, it is clear that there is a high %age of use of the methods for solving these challenges. But it shows the highest %age in dimension reduction and classification of gene expression data and protein structure prediction. The researchers have gained great success in solving the challenge of protein structure prediction, i.e., they have got 100% accuracy in protein structure prediction.

Figure 1.6: Comparative views of utilization of integrated KBMs and CIMs for biological tasks in Proteomics



From the study, it is also observed that the methods are highly employed for the dimensionality reduction and classification of cancer/tumor datasets. But none of the application has been found for the dimension reduction and classification of NMDs. So, in our work, we present novel integrated methods for the dimension reduction and classification of both binary and multi-class NMD disorder data sets.

From the literature, it is also observed that SVM is the most widely used classification methods for the classification of data sets. As in literature of gene selection and classification of disease data sets, SVM is the highly deployed method for the classification, as it can be seen in [16], [54]–[56], [66], [71], [76], [77], [115], [123], [124], [129], [131], [145]. SVM also offers assorted advantages like ability to condense information, absence of local minima, good generalization capacity, high prediction ability, scalability, fast convergence, reliable prediction and robustness in the noisy environment, high accuracy, specificity and sensitivity [51] [53] [58]. The problem reported with SVM is that it does not allow for knowledge extraction and automatic feature selection and provides low comprehensibility [53]. So, in order to deal with high dimensional data, we need to first reduce the dimension of the data set by incorporating any one of the mechanism of dimensionality reduction, then the classification of data will be done.

1.6 Conclusions

From the review, it is observed that the CIMs are extensively employed in bioinformatics. As compared to them, the applications of KBMs are least found because the computational intelligence methods are highly advantageous in solving the problems. The individual methods like ANN and EA are broadly employed, whereas DM is moderately used; and FL and SI are less used methods. The integrated methods, namely ANN-EA, DM-EA and EA-FL are also very widely applied in bioinformatics. The problems in these areas need to be resolved in different phases, such as the error occurred in the first phase is cumulated in second phase such as if proper feature selection method is not used, then error will be cumulated in classification, the error occurred in the conversion of sequence into multiple alignment or pairwise alignment, will be cumulated in encoding the sequence, the error encountered in filtering and classification steps will be cumulated in the extraction of the rules. Therefore, it is desired for an accurate prediction/classification that the errors should be removed at the initial phases itself. Otherwise the result will get the cumulative error at all the steps (from beginning to end). This study will help the novice researchers to choose appropriate KBMs and CIMs to deal with the challenges occurred in the representation, integration, analysis, interpretation and management of biological data.

1.7 Motivation

According to the report of "The Cooperative International Neuromuscular Research Group", duchene muscular dystrophy occurs in 1/3500 people, spinal muscular atrophy occurs in 1/6000 people, cerebral palsy occurs in 6/1000 male births, myotonic muscular dystrophy occurs in 1/8,000 people, myasthenia gravis occurs in 1/10,000 people, becker muscular dystrophy occurs in 1/18,450 people, facioscapulohumeral muscular dystrophy occurs in 1/20,000 people, amyotrophic lateral sclerosis occurs in 7/100,000 people and more than 400,000 individuals are affected by cerebrovascular disorders. The earlier methods of diagnosis are highly complex, which leads to a great level of difficulty in understanding the severity of disease occurred. These disorders follow a heterogeneous pattern of pathogenesis. So, in order to correctly diagnose a disease, we need to consider the gene expression levels through microarray experiment. But the extremely high dimension of NMD gene expression data sets poses a great challenge in successful diagnosis of the diseases. So, the need of monitoring of gene expression levels in the occurrence of a disease has motivated the development of an intelligent integrated method which first reduces the dimension of the data set by selecting the most important genes for diagnosis of the disease. The developed intelligent integrated method takes into account various parameters which show its success, such as accuracy, sensitivity, specificity, positive predicted value and negative predicted value. This thesis develops an integrated intelligent method with the ultimate aim of helping biologists in gene selection and accurate classification of various NMD disorders.

1.8 Objectives of Thesis

As observed from the literature review, the high throughput microarray technologies lead to the complex high dimensional and noisy gene expression data, and a limited number of observations as compared to the large number of gene expression values. These characteristics badly affect the analysis of microarray datasets and pose a challenge for building an efficient diagnostic model. Hence, there is a critical need to apply data-mining and computational intelligence methods to analyze these datasets efficiently. Therefore, the objectives of our thesis are

1) To apply computational intelligence method to remove noisy and redundant genes.
2) To apply computational intelligence method for extraction of discriminative genes.
3) To apply computational intelligence method for classification of gene expression data.

Steps to achieve objectives:

Step 1: To study all the individual CIMs applied to the microarray gene expression data for dimensionality reduction.

Step 3: To study all the integrated CIMs applied to the gene selection from gene expression data sets.

Step 4: To study all the methods applied for the diagnosis or classification of samples of diseases.

Step 5: To develop a new integrated method for preprocessing of data, for dimensionality reduction of data, for the classification of gene expression data and thus for the diagnosis of disease.
Step 6: To implement the proposed method on the publicly available data sets.
Step 7: To calculate the performance measures used to evaluate the quality of the proposed model.

1.9 Outline of the Thesis

The thesis is structured into eight chapters, in which the first chapter is all about the introduction and a comprehensive review of literature. The literature review is divided into two parts, i.e., individual methods and integrated methods. The individual methods are artificial neural network (ANN), fuzzy logic (FL), etc. The integrated methods are artificial neural network-swarm intelligence (ANN-SI), artificial neural network-evolutionary algorithm-random forest (ANN-EA-RF). The %age employment of all the methods is shown graphically in each area considered. Then it contains the motivation of the work and objectives of the thesis. At the end of chapter 1, a brief plan of all the chapters in the thesis is given.

In chapter 2, we provide all the basic concepts used in this thesis. It includes the basic biological background information and problem statement, introduction to gene expression data, microarray technology, neuromuscular disorder classification problem and the associated issues, publicly available neuromuscular disorder data sets, feature selection and methods, classification and its methods, model selection parameters and model validation techniques.

In chapter 3, we describe the proposed methodology “Diagnosis of facioscapulohumeral muscular dystrophy using cosine distance metric based hierarchical clustering and k-nearest neighbor”. In this, firstly cosine distance metric-hierarchical clustering method is applied to cluster the genes in the data set. The K nearest neighbor method is used to classify the samples using the clustered genes. The model proposed is cosine distance metric based hierarchical clustering-KNN and is compared with k-means-LDA, k-means-QDA, k-means-KNN, euclidean distance metric based hierarchical clustering-LDA, euclidean distance metric based hierarchical clustering-QDA, euclidean distance metric based hierarchical clustering-KNN, cosine distance metric based hierarchical clustering-LDA, cosine distance metric based hierarchical clustering-QDA. The proposed model results are compared with other models in terms of accuracy, sensitivity, specificity, positive predicted value and negative predicted value.

Chapter 4 is about "An integrated algorithm for dimension reduction and classification applied to microarray data of neuromuscular dystrophies". In this chapter, supervised techniques are applied for dimensionality reduction. First, entropy filter feature selection method is employed to rank and sort the genes according to their importance. The highly discriminating genes are selected from all the genes. Then classification of samples is performed using the linear SVM by using only the selected genes. The proposed model entropy-linear SVM is compared with t-test-KNN, t-test-linear SVM, entropy-KNN. Again, the results are compared in terms of accuracy, sensitivity, specificity, positive predicted value and negative predicted value.

Chapter 5 presents the proposed methodology "Building an intelligent integrated method of gene selection for facioscapulohumeral muscular dystrophy diagnosis". In this chapter, the task of feature selection is done in three phases where the gene selection is done in the first two phases. In the first phase, a filter method, t-test is employed for the preselection of genes. In the second phase, an embedded method, genetic algorithm is deployed for the selection of genes. The first phase and second phase collectively selects the genes. Then, using those selected genes, the classification of samples is done using KNN. The proposed model, i.e., t-test-genetic algorithm-KNN is compared with t-test-genetic algorithm-LDA and t-test-genetic algorithm-QDA. Here also, the results are compared using the same parameters, i.e., accuracy, sensitivity, specificity, positive predicted value and negative predicted value.

Chapter 6 discusses the proposed methodology "A novel hybrid feature selection model for classification of NMDs using Bhattacharyya coefficient, genetic algorithm and radial basis function based support vector machine". Here also, the task of feature selection is done in two phases. In the first phase, the Bhattacharyya coefficient is employed for gene ranking and preselection. In the second phase, the genetic algorithm is used for selection of genes. Then at the last, classification is done using radial basis function support vector machine (RBF SVM) using the only selected genes after the second phase of gene selection.

Other classification methods applied after the second phase are LDA, QDA, KNN and linear SVM. The performance parameter taken for comparison is the accuracy.

Chapter 7 presents the proposed methodology “A novel approach for gene selection and multi-class classification of neuromuscular disorders”. In this methodology, the task of gene selection is done using a median matrix which is created after processing gene expression matrix. Highly compact subsets of genes are selected using the matrix and these genes are used for the classification of samples using the RBF SVM method. Few other classification methods are also used for classification of samples, i.e., LDA, QDA, KNN and linear SVM. The model selection parameter is again accuracy. Leave-one-out cross validation technique is used to validate the model.

Chapter 8 presents the conclusion of the all the chapters of the thesis. The future work is also given at the end of this chapter.

The detailed literature review presented in this chapter has been published in International Journal of Computational Biology and Drug Design, Vol. 9, No. 3, pp. 173-227, 2017, Inderscience Publishers, United kingdom, DOI: 10.1504/IJCBDD.2016.078277, mentioned under list of publications after chapter 8.

Chapter 2

Basic Concepts

In this chapter, we give information about the biological background, the problem statement, microarray data sets; various feature selection methods, classification methods, performance measures and the model evaluation methods. Section 2.2 gives the detailed information about the background, the problem statement and the various issues related to it. Section 2.3 details all the NMDs data sets used in this thesis. Section 2.4 introduces the feature selection methods and its categorization. Section 2.5 gives the various classification algorithms employed to classify the samples using to their respective classes. The performance measures employed to access the performance of various models and a selection of models is given in section 2.6. Section 2.7 gives a variety of the model validation techniques used in the work to validate the model.

2.1 Introduction

The diagnosis of NMDs is an active research area in bioinformatics. The correct classification of NMDs has great impact in providing the best treatment options to the patients. Earlier the diseases were diagnosed by considering the morphological appearances of the diseases. But these types of diagnostic methods have several restrictions in their diagnostic capability. The particularity of treatment specific to the kind of NMD distinguished by the pathogenic blueprint may increase the efficiency of the patients. To better understand the NMD classification problem, the diagnostic approaches based on the analysis of gene expression data have been given. As, the gene expression level is known to solve the fundamental issues relating to the prevention, diagnosis, biological evolution mechanism and drug discovery for a cure. The modern progress in microarray technology has provided us a way to monitor thousands of genes simultaneously, which affected the development in NMD classification procedures using gene expression data.

The various issues related to the gene expression data are: first, the gene expression data usually have very high dimension, i.e., tens of thousands of genes; second, the number of samples is very less and third most of the genes in the dataset are irrelevant to the NMD classification. Thus, in order to accurately classify the NMD dataset, we need to reduce the dimension of the dataset by removing the irrelevant genes from it. The dimensionality reduction techniques were introduced to remove out the noise, irrelevant and redundant features from the dataset. The several goals of dimensionality reduction techniques are lowering the computational complexity of the system, improving the learning performance, building the better generalization model, decreasing the cost of classification and the required storage. Some researchers proposed to reduce the dimension of the NMD dataset before the classification task. It helps to remove out the noise, redundant and irrelevant genes which further increases the classification accuracy and reduces the running time.

2.2 Background Information and Problem Statement

2.2.1 Biological background information

During the process of reproduction, individuals transmit some particles known as genotype to their offspring. The public display of the genotype is called the phenotype. In every living system, cells are the fundamental working units. Of the two different types of cells, each organism is made up of one type, i.e., prokaryotic cells and eukaryotic cells. Inside the eukaryotic cells, there is a nucleus. The cells contain proteins whose behavior, concentration and shape represents the attributes of a cell. For example, fat cells and hair cells are varied because they are consisting of different proteins. The instructions required to direct the activity of a cell are contained within the deoxyribonucleic acid (DNA). The ribonucleic acid (RNA) works as a mediator for these activities.

DNA is the genetic material which is transmitted over generations. It is also known as the blueprint of all living organisms because the genetic information needed for building and maintaining the life in a cell is encoded by DNA. The DNA holds the information on how a cell works. The organization of a DNA molecule consists of a double helix structure which includes phosphate group, sugar molecule and a nitrogen base. The bases are adenine (A), guanine (G), cytosine (C) and thymine (T). Hence, every nucleotide is made up of a deoxyribose sugar, a phosphate group and one of the four nitrogen bases. The arrangement of base pairs in a DNA strand actually tells the instructions required to do a particular activity. The genome encodes for the whole DNA sequence that code for a living thing. The full set of DNA in an organism is called the genome. The genome breaks down into a set of genes where each and every gene is defined for some unique and specific purpose. The proteins become active in the presence of a third molecule, i.e., RNA whose structure is similar to DNA. A chromosome is the configuration where the genetic material is organized into several separated DNA molecules.

2.2.2 Gene expression

The formulation of an organism and the properties of cells are determined by the sequence of four nucleotides which are tied along the DNA chain. Genes which further produce proteins are represented by the sequence of nucleotides in the DNA chain. The expression of genetic information stored in the DNA molecule occurs in two phases: (i) transcription stage (ii) translation stage. In the transcription phase, the mRNA is transcribed from the DNA molecule and in the translation stage, amino acid sequences of proteins are translated from mRNA that performs various cellular functions. At a given time, a gene can be highly active, moderately active or inactive. The activity level of a gene is used to indicate the rate at which the corresponding protein is produced by means of RNA. The gene expression is defined as the procedure of transcribing a gene's DNA sequence into RNA. The level of gene expression shows the amount of mRNA produced during the process of transcription. The various biological states such as embryogenesis, cell development, etc. are associated with the occurrence of specific patterns of gene expression data. So, the gene expression level indicates the activity or functionality of a gene under some biochemical conditions.

When some changes occur in the DNA molecules during the process of replication, they cause some serious diseases in the human body. The mutation or error in a gene(s) changes the normal cells to malignant cells when replicating, which ultimately causes the disease. This mutation can be due to the substitution, insertion, deletion, inversion or recombination in the DNA molecule. The substitution mutation occurs when a nucleotide change into another nucleotide, e.g. from C to A or from C to A. During inversion mutation, the nucleotide inverts by 180 degrees. Another type of mutation, i.e., recombination mutation influences part of nucleotides between homologous sequences of homologous chromosomes. Another type of mutation is insertion mutation in which a long sequence of nucleotide is inserted into a DNA sequence. In deletion mutation, a long sequence of nucleotide is deleted from a DNA sequence. If a mutation in a DNA sequence does not affect the process means the production of amino acids after the mutation is same as amino acid before the mutation, then the mutation is known as silent or synonymous. If both are not same, then it is known as non-synonymous. The gene expression analysis involves examining the gene expression levels of a large number of genes simultaneously under the condition of interest. The microarray technology is effectively used to examine a large number of genes of an organism simultaneously under some condition.

2.2.3 Microarray technology

The microarray technology is used to monitor the expression level of genes of a whole organism simultaneously under the given conditions of interest, which is called the gene expression analysis or the gene expression profiling. There are two technologies which analyze the entire genome simultaneously (i) microarrays (ii) Serial analysis of gene expression (SAGE). The microarrays are most popular technology which is further of two types: complementary DNA microarrays and high density oligonucleotide arrays which compute the relative level of mRNA produced between different samples. The SAGE technology measures the absolute level of mRNA produced between different samples. The most common application of microarray technology is to compare the gene expression set maintained in two conditions, i.e., normal condition and a particular condition, e.g. to compare the expression profiles for diseased cells and normal cells to identify the diseased genes and to discover the drug by comparing these expression profiles.

The expression ratio of a gene represents the expression differences between two conditions. Usually normalization is done to remove out the variability of cDNA microarray data and to process the data which can then be shown in the structure of gene expression matrix.

2.2.4 Neuromuscular disorder classification problem

The NMD classification problem is defined as to identify to which of the class labels, a new sample belongs based on the training data set. This problem is addressed by employing gene expression profiling. Various studies have revealed that the diseases are related to some mutations in genes or change in their expression levels. The present work solves the two types of classification problems, i.e., gene selection for classification of binary class datasets and gene selection for classification of multi-class datasets. There is no single classification algorithm that is superior over the rest. Some classification algorithms work only on the binary classes and are not extensible to the multi-class problem, while some classification algorithms are more broad and flexible. The problem of classification of NMDs using gene expression levels is different from other classification problems due to its distinctive nature and the application area.

2.2.5 Various issues

There are a few issues related to the challenge of classification of microarray gene expression datasets. The first issue is arrived due to the uniqueness in gene expression data sets. The sample size of publicly available gene expression datasets is very small. But on the other hand, with the help of microarray technology, it is now doable to monitor the dataset containing tens of thousands of genes. This nasty situation of small sample size and the high dimensional gene dataset is a very big challenge for classification algorithms for NMD classification. Most of the classification algorithms were not developed by keeping these features of data set in the mind. So, most of the dataset with such a high dimension and sparse samples tends to overfit which is really a very big issue. The huge number of genes in a sample increases the computational overhead and training time.

The second issue with the classification of a NMD dataset is the presence of noise due to the biological and logical reasons. The noise arises due to biological reasons causes problems because it is found that most of the genes in the dataset are not relevant for the predicting the classes of NMDs. The logical noise occurs during the processing of data at different stages. Due to these kinds of noise in the data, it is extremely difficult to do the proper classification of NMDs.

The third issue is the existence of a huge number of irrelevant genes in the dataset. Most of the genes in the dataset are not particularly related to the class of NMD; hence a large number of genes are irrelevant to the class. The discriminatory power of the most relevant genes gets decreased due to the existence of a huge number of irrelevant genes. It also increases the computation time while increasing the difficulty in the classification.

So to avoid these nasty issues, we need to select a few genes which lead to the accurate classification. This would help in making an efficient and accurate classification algorithm by using those genes in the training phase.

2.3 Publicly Available Microarray Neuromuscular Disorder Data Sets

The NMD data sets are taken from Gene Expression Omnibus (GEO) and National Centre for Biotechnology Information (NCBI).

2.3.1 DST-1

The first dataset (named as DST-1) taken is of Facioscapulohumeral muscular dystrophy (FSHD) dataset entitled “Transcriptional profiling in facioscapulohumeral muscular dystrophy to identify candidate biomarkers” with accession id GSE36398. The data set contains the expression level of 50 samples and 33,297 genes in which 24 samples are healthy and 26 samples are affected by FSHD. The dataset contains the samples of RNA extracted from biceps and deltoids. The experiment was run on the platform GPL6244 Affymetrix Human Gene 1.0 ST Array [152].

2.3.2 DST-2

The second dataset (named as DST-2) taken is of again FSHD from NCBI. The data is a small part of the accession id E-GEOD-3307 entitled “Transcriptional profiling by array of 12 human muscle diseases”. The experiment was run on arrays A-AFFY-33 – Affymetrix GeneChip Human Genome HG-U133A and A-AFFY-34 - Affymetrix GeneChip Human Genome HG-U133B. The dataset taken for this experiment was run on A-AFFY-33 – Affymetrix GeneChip Human Genome HG-U133A. The complete dataset contains the 13 data sets of profiling of human skeletal muscles and a total of 242 samples. From the whole dataset, the small part of the data is taken which consists of 14 samples affected by FSHD and 18 normal samples. The dataset contains the expression level of 22,645 genes [153].

2.3.3 DST-3

The third dataset (named as DST-3) is of Juvenile Dermatomyositis (JDM) with accession id E-GEOD-3307 entitled “Transcriptional profiling by array of 12 human muscle diseases”. It contains 21 samples affected by JDM and 18 healthy or control samples. Again, the whole dataset contains the expression level of 22,645 genes [153].

2.3.4 DST-4

The fourth dataset (named as DST-4) is again from E-GEOD-3307. This dataset differentiates between Amyotrophic Lateral Sclerosis (ALS), Fascioscapulohumeral Muscular Dystrophy (FSHD), Juvenile Dermatomyositis (JD), Duchenne Muscular Dystrophy (DMD) and healthy (normal) samples. It involves a total of 72 samples, where 9 samples belong to ALS, 14 samples to FSHD, 21 samples to JD, 10 samples to DMD and rest 18 samples to healthy class. The expression levels of 22,645 features were monitored in each sample [153].

2.3.5 DST-5

The fifth dataset (named as DST-5) involves Acute Quadriplegic Myopathy (AQM), Becker Muscular Dystrophy (BMD), Limb Girdle Muscular Dystrophy 2A (LGMD-2A), Limb Girdle Muscular Dystrophy 2B (LGMD-2B), Limb Girdle Muscular Dystrophy 2I (LGMD-2I) and healthy samples. The dataset contains total 55 samples which include 5 samples of AQM, 5 samples of BMD, 10 samples of LGMD-2A, 10 samples of LGMD-2B, 7 samples of LGMD-2I and 18 healthy samples. In this dataset, a total of 22,283 features were represented in each sample [153].

2.3.6 DST-6

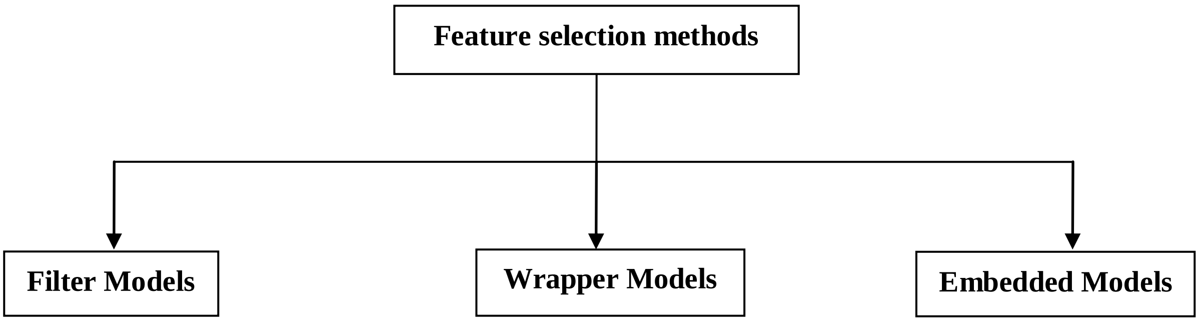
The sixth dataset (named as DST-6) was taken from EMBL-EBI which differentiates between various kinds of muscular dystrophies; there are total 121 RNA samples of 13 classes of human skeletal muscles under the experiment array. The classes are acute quadriplegic myopathy (AQM), amyotophic lateral sclerosis (ALS), becker muscular dystrophy (BMD), facioscapulohumeral muscular dystrophy (FSHD), juvenile dermatomyositis (JD), duchene muscular dystrophy (DMD), hereditary spastic paraplegia (SPG4), autosomal dominant Emery-Dreifuss muscular dystrophy (AD-EDMD), X-linked recessive Emery-Dreifuss muscular dystrophy (X-Linked-EDMD), limb girdle muscular dystrophy 2 A (Caplain-3), limb girdle muscular dystrophy 2 B (Dysferlin), limb girdle muscular dystrophy 2 I (FKRP) and normal human skeletal muscles (NHSM) which have 5, 9, 5, 14, 21, 10, 4, 5, 3, 10, 10, 7 and 18 samples respectively. The expression levels of 22,645 genes were reported in each sample in the entire dataset [153].

2.4 Feature Selection

There are two types of dimensionality reduction methods, i.e., feature extraction and feature selection. Feature extraction methods transform the data into new feature space with lower dimensionality by combining the original features to form new constructed features. The various examples of feature extraction approaches are PCA, LDA, and canonical correlation analysis. Feature selection methods, particularly selects a small feature subset that minimizes redundancy and maximize significance to the class labels in classification. The feature selection techniques are considered better than the feature extraction techniques because feature extraction techniques transform the original feature space to a new feature space with lower dimension by combining the original features. It is extremely difficult to link the features from original feature space to new transformed feature space and the further analysis of new features is problematic since the transformed features bear no physical meaning. But feature selection techniques select a subset of features from the original feature set without any transformation which preserves the physical mapping and the meaning of the original features. So the features selected using feature selection techniques are more interpretable and more readable.

For the task of classification, the feature selection techniques should select only those features which are capable of discriminating samples that belong to different classes. The feature selection techniques aim to select a feature subset which avoids the issue of overfitting, reduces the cost of classification; provides higher classification accuracy, classifier independence, compactness and biological relevance. Besides these benefits, feature selection techniques also add an extra layer of complication to the system. The feature selection algorithms can serve to both supervised and unsupervised learning. Supervised learning is applied when the class labels are known a-priori. The unsupervised feature selection models, search for the feature subset when the class labels are unknown, always depending upon the clustering quality measures. The supervised feature selection techniques are considered better because they assesses the relevance of features guided by the label information, but a good selector needs enough data which are very time consuming. While the unsupervised feature selection works with unlabeled data, so it is very difficult to assess the significance of features. The supervised feature selection techniques can be further categorized into three types, focusing on how they embed the feature selection search with the classification model shown in figure 2.1.

Figure 2.1: Categorization of feature selection methods



Filter models

The filter model assesses the significance of features by taking a glance only at the intrinsic characteristics of the data and without utilizing any particular classification algorithms. The general features of the data are distance, consistency, information, correlation and dependency. The feature selection process is separated from the classifier learning so that the bias of a learning algorithm does not interact with the bias of a feature selection algorithm. A filter model is usually made up of two steps: first a relevance score is calculated based on some feature evaluation criteria. This feature evaluation can be either univariate or multivariate. In the former way of feature evaluation, each feature is ranked independent of the other features without considering the correlation among genes and in the latter way of feature evaluation, features are ranked while considering the correlation among them. Second, the features with the high relevance score are given as input to the classification method and the rest of the features are discarded. The various advantages of using univariate filter feature selection methods are they are fast, independent of any specific classifier and are easily scalable. Besides these advantages, univariate filter methods ignore the feature dependencies and the interactions with the classifier. But on the other hand, the multivariate feature selection model benefits us by considering the feature dependencies, and they are also independent of the classifier. The disadvantages are time-consumption and less scalability in comparison of the univariate models. Likewise, they also pay no attention to the interaction with the classifier. The filter models are usually applied as a pre-processing step in feature selection for classification.

Wrapper models

Wrapper models use the classification accuracy of a specific learning algorithm to calculate the excellence of chosen features. They choose a feature subset by evaluating a specific classifier and they keep on searching and evaluating the subsets of features until the desired quality is achieved. The architecture of a wrapper model of feature selection consists of two main steps: first it contains a feature selection search component which searches the subsets of features from all the possible feature subsets. Secondly, it has a feature evaluation component which evaluates the subsets of features utilizing a specific classifier. In the next iteration, it goes back to the feature search component for the next feature subset for which the quality is estimated using that particular classifier. They keep on iterating until the desired quality is reached. The subset of features with highest accuracy is chosen as the final subset to learn the classifier. A number of search strategies can be used: hill climbing, best-first, branch-and-bound and GA. These models often obtain better classification accuracies as compared to filter models. But these models are very expensive to run for data with a large number of features, highly computationally expensive, often prone to overfitting and they can perform only classifier dependent selection.

Embedded models

These models were introduced to bridge the gap between filter models and wrapper models. The embedded models of feature selection, selects a subset of features while incorporating the feature selection procedure in the learning algorithm itself. It takes the advantages of both filter models and wrapper models. First, it incorporates the statistical criteria as filter models to select the several candidate feature subsets with a given cardinality. Second, it chooses the subset with highest classification accuracy. So the embedded models are comparatively efficient as filter models and comparatively accurate as wrapper model. But as compared to wrapper model, these embedded models are less computationally expensive. These models achieve both model fitting and feature selection simultaneously.

2.5 Classification

For the classification of samples, different linear and nonlinear classification methods are used. From the former category, LDA is used and from the latter category, QDA, KNN, SVM are used to classify the samples.

2.5.1 Linear discriminant analysis

LDA was developed by R.A. Fisher in 1936. It is a joint probability model. It is known as the generalization of Fisher's linear discriminant [154]. It is a linear classifier used to locate the linear combination of features to separate the objects of two or more classes. LDA works on the concept of covariance matrices which tries to provide more class seperability by drawing a decision region between the given classes. It assumes that the covariance of each of the classes is identical. Here, it uses the "class-independent-transformation" approach which considers each class as a separate class against all the other classes. It maximizes the ratio of the overall variance of the within class variances which guarantees the maximal seperability. This technique can also be used as a preprocessing step in pattern classification and dimensionality reduction. The benefit of LDA is its simplicity and it requires no parameter tuning.

LDA works as follows: Compute the mean of the entire data set and the both data sets. Let μ_1 be the mean of the entire data set, μ_2 and μ_3 denotes the mean of dataset1 and dataset2 respectively. The mean of the entire data set and within class scatter is computed in equations 2.1 and 2.2 respectively. For a two class classification problem, the apriori probabilities are assumed to be 0.5, so the within class scatter is shown in equation 2.3. The between class scatter and the optimizing criteria are shown in equations 2.4 and 2.5 respectively.

$$\mu_1 = p_1 * \mu_2 + p_2 * \mu_3 \tag{2.1}$$

$$S = \sum_j p_j * (cov_j) \tag{2.2}$$

$$S_w = 0.5 * cov_1 + 0.5 * cov_2 \tag{2.3}$$

$$S_b = \sum_j (\mu_j - \mu_1) * (\mu_j - \mu_1)^T \tag{2.4}$$

$$criterion = inv(S_w) * S_b \tag{2.5}$$

2.5.2 Quadratic discriminant analysis

QDA is a generalized version of LDA. It is a nonlinear classifier used to assign data to one or more classes. The possibility of an observation being in a sample is determined by the quadric surface. A new sample is classified into that class which has the smallest squared distance. Like LDA, it is useful when class labels are known beforehand. QDA is known to have more predictability power than LDA. Unlike LDA, there is no assumption that the covariance of each of the classes is identical.

2.5.3 K-nearest neighbor

KNN is a non-parametric method which can be employed for both regression as well as classification tasks [156]. It is a type of supervised and lazy learning [155], and is a non-generalizing method. KNN is also known as an instance based learner which works on the "similarity based approach" or "learning by analogy". It is based on the minimum distance measure between the test samples and training samples. KNN is a bit more computationally expensive as compared to other learners. They need high storage requirement and efficient algorithms to process them.

The input to the algorithm is given in the form of vectors as a training data set with their selectors which must reside in the memory at the run time. It stores all the training cases in memory, based on which it classifies the testing cases by searching k-most similar cases from training data. In lazy learning, a classifier in its learning phase does not construct any model for the classification of any test tuple. It simply stores the training tuples in the memory. When it encounters the test tuple for classification, it performs generalization and constructs a model so that it can classify the test tuples based on the similarity of the stored training tuples. The lazy learners perform a huge amount of work when presenting the classification of test tuples. That is why it is called memory-based classification. The tuples will be defined as the "nearest neighbors" of the unknown tuple. The word "nearest" is defined in the term of "closeness" which is a distance metric. The various distance metrics are "euclidean", "cosine", "hamming", "Manhattan" and "correlation".

2.5.4 Support vector machine

SVM was introduced in COLT-92 by Boser, Guyon and Vapnik [157]. These are a set of supervised learning algorithm used for both classification as well as regression tasks. It is used to classify both linear and non-linear data. SVM uses two notions to solve any problem, i.e., large-margin separation and kernel functions. SVM transforms the original data to higher dimension by the means of non-linear mapping. It searches for a decision boundary, i.e., a linear separating hyperplane in the higher dimension which differentiates the tuples of one class from another class.

For binary class classification problem where the classes are linearly separable, let us assume that we have D given in equation 2.6. In equation 2.6, X_i is the set of training tuples with associated class labelsy_i. The value of y_i can be either of these two, +1 or -1, which corresponds to the classes $result = diseased$ and $result = healthy$ respectively. There are a number of separating hyperplanes shown by the dashed lines. The task here is to find the separating hyperplane which gives the minimum classification error for the classification of test data, that hyperplane will be considered as the "best" hyperplane. The maximum marginal hyperplane is necessary for accurately classifying the data. We assume that the hyperplane with the maximum margin is more accurate than the hyperplanes with smaller margins for classifying the tuples of test data. That is why, SVM always search for the maximum marginal hyperplane. A separating hyperplane is defined in equation 2.7, W is the weight vector and b is bias. The tuples in the training data are 2-D where x_1 and x_2 are the values of attributes A_1 and A_2 respectively. If b is considered as an additional weight than the equation 2.7 is rewritten as equation 2.8. Any point which lies above the separating hyperplane should satisfy the equation 2.9 and any point which lies below the separating hyperplane should satisfy the equation 2.10. The hyperplanes which defines the sides of the margin are given in equations 2.11 and 2.12. Any tuples in the test data set which lies on or above H_1 is associated with the class +1 and any tuples which lies on or below H_2 is associated with class -1. Joining the equations 2.11 and 2.12, we get equation 2.13.

$$(\mathrm{X}_1, \mathrm{y}_1), (\mathrm{X}_2, \mathrm{y}_2) \cdots\cdots (\mathrm{X}_{|\mathrm{D}|}, \mathrm{y}_{|\mathrm{D}|}) \tag{2.6}$$

$$\mathrm{W.X} + \mathrm{b} = 0 \tag{2.7}$$

$$\mathrm{w}_0 + \mathrm{w}_1\mathrm{x}_1 + \mathrm{w}_2\mathrm{x}_2 = 0 \tag{2.8}$$

$$\mathrm{w}_0 + \mathrm{w}_1\mathrm{x}_1 + \mathrm{w}_2\mathrm{x}_2 > 0 \tag{2.9}$$

$$\mathrm{w}_0 + \mathrm{w}_1\mathrm{x}_1 + \mathrm{w}_2\mathrm{x}_2 < 0 \tag{2.10}$$

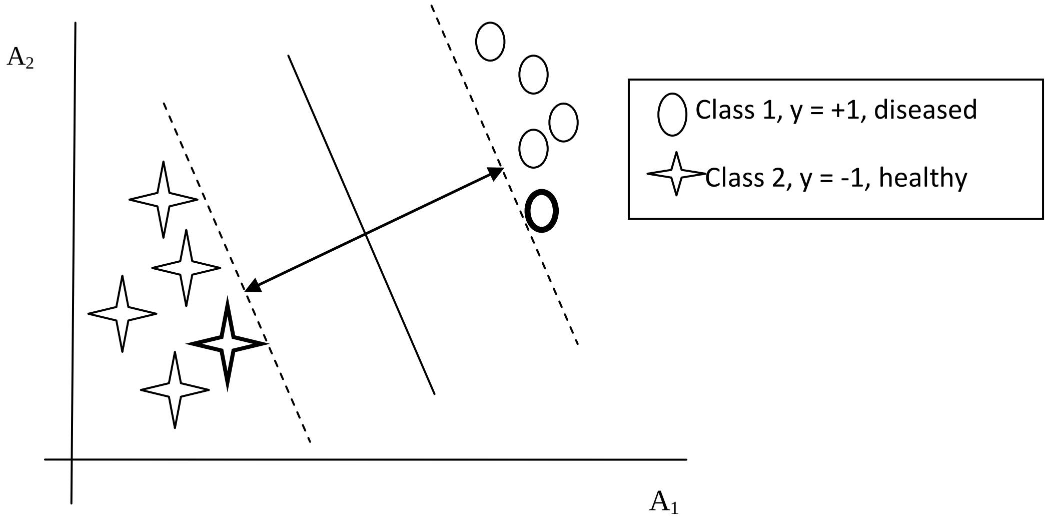
$$H_1 : w_0 + w_1x_1 + w_2x_2 \geq 1 \text{ for } y_1 = +1 \tag{2.11}$$

$$H_2 : w_0 + w_1x_1 + w_2x_2 \leq -1 \text{ for } y_1 = -1 \tag{2.12}$$

$$y_1(w_0 + w_1x_1 + w_2x_2) \geq 1, \text{for all i} \tag{2.13}$$

The support vectors are the data points or tuples which lie on H_1 or H_2 satisfy the above equations. The vectors give the most important information for classification and are very difficult to find. These are shown in figure 2.2. In figure 2.2, the support vectors are shown with the darker color. We can also calculate the size of the maximal margin. The distance from the separating hyperplane to any point on H_1 is $\frac{1}{\|W\|}$ where $\|W\|$the euclidean norm of W, i.e., is$\sqrt{w.w^2}$. So, the maximal margin is$\frac{2}{\|w\|}$. The linear SVM classifier is defined as the inner product between two vectors, which is given in equation 2.14. The decision function f(y) decides how to classify the data and assigns a score for the input y. The decision function for linear classifier is of the form of equation 2.15. Here z is weight vector and b is bias. It should satisfy the set of inequalities given in equation 2.16. The RBF kernel is given in equation 2.17.

Figure 2.2**:** Support vectors



$$\langle z, y\rangle = \sum_{j=1}^{M} z_j y_j \tag{2.14}$$

$$f(y) = \langle z, y \rangle + b \tag{2.15}$$

$$f(x) = \begin{cases} > 0, & y_j \in c_1 \\ < 0, & y_j \in c_2 \end{cases} \tag{2.16}$$

$$K(d, d_i) = exp(-\gamma||d - d_i||^2) \tag{2.17}$$

2.6 Model Selection

Model selection is defined as the process of choosing one "best" classifier from all classifiers if we have more than one classifier. There are some metrics which help in model selection by evaluating the performance of classifiers.

2.6.1 Metrics for model selection

The various metrics for model selection or evaluating the performance of a classifier are: accuracy (ACC), sensitivity (SNS), specificity (SPC), positive predicted value (PPV) and negative predicted value (NPV). For the problem of classification, we take two classes, i.e., positive class and negative class. Let us suppose, there are two classes of any disease diagnostic problem, namely diseased class and healthy class. According to the condition of interest, the samples in positive class are diagnostic_result = positive, while the samples in negative class are diagnostic_result = negative. Suppose we have trained our classifiers on the training data and now we are testing the classifier's prediction ability on the test set. Here are some terminologies which we are using: P defines the numeral of samples in the positive class and N represents the numeral of samples in the negative class.

The four building blocks which are needed for the evaluation of a classifier's class prediction ability are true positive (TP), true negative (TN), false positive (FP) and false negative (FN). TP is defined as the numeral representing the positive samples which are correctly classified by the classifier. TN is defined as the number of negative samples which are correctly classified by the classifier. FP is the number of negative samples which are incorrectly classified as positive. The actual class for these samples is diagnostic_result = negative, but the predicted class by the classifier is diagnostic_result = positive. FN is the number of positive samples which are incorrectly classified as negative. The actual class for these samples is diagnostic_result = positive, but the predicted class by the classifier is diagnostic_result = negative. All these measures are placed in a useful tool which analyses the classifier’s performance, confusion matrix. It is a table which represents the number of correct and number of incorrect predictions. P’ defines the total number of samples either correctly or incorrectly classified as positive. N’ defines the total number of samples either correctly or incorrectly classified as negative.

Table 2.1: Confusion matrix

Predicted class

		Yes	**No**
Actual class	**Yes**	TP	FN
	No	FP	TN
	Total	P’	N’

The total number of samples in the dataset is calculated by the equation 2.18

$$S = TP + TN + FN + FP \tag{2.18}$$

The performance metrics are calculated from the confusion matrix. ACC is defined as the classifier’s ability to accurately classify the samples. It represents the percentage of samples that are correctly classified by the classification algorithms. It is given in equation 2.19. The misclassification rate (MCR) is also calculated which is shown in equation 2.20. SNS represents the classifier’s ability to correctly predict the positive samples. It is the ratio of true positive and total number of samples and is given in equation 2.21. SPC defines the classifier’s ability to correctly predict the negative samples. It is the ratio of true negatives and total number of samples and is given in equation 2.22. PPV represents the ratio of true positives and the total number of samples predicted to be true either correctly or incorrectly. It is also called precision and defined in equation 2.23. NPV represents the ratio of false positives and the total number of samples predicted to be false either correctly or incorrectly. It is defined in equation 2.24.

$$ACC = \frac{TP + TN}{S} \tag{2.19}$$

$$MCR = 1 - ACC \tag{2.20}$$

$$SNS = \frac{TP}{S} \tag{2.21}$$

$$SPC = \frac{TN}{S} \tag{2.22}$$

$$PPV = \frac{TP}{TP + FP} \tag{2.23}$$

$$NPV = \frac{TN}{TN + FN} \tag{2.24}$$

Besides these performance metrics, there are a few other aspects on which the performance of a classifier is evaluated. These are: robustness, scalability and speed. Robustness is defined as the classifier's ability to correct predicts the samples given the data contain missing value and noise. A classifier is said to be robust if the performance of a classifier does not degrade when dealing with the noisy data. Scalability represents the classifier's ability to correctly predict the samples given a huge dimensional data. A classifier is said to be scalable if its performance does not get reduced when dealing with large amount of data. Speed is defined using the term computational cost involved when classifying the data.

2.7 Model Validation

2.7.1 Holdout validation

In holdout validation, the dataset is divided into two parts: training data set and test data set. According to the conventional rule of validation, the dataset is divided in the ratio of 70-30, i.e., 70 % training data set and 30 % test data set. The training dataset is used by the function approximator to fit the function and derive the model. The function approximator predicts the output values for the data in the testing dataset.

2.7.2 Cross validation

It is a model validation technique for evaluating how the results of a statistical analysis will generalize to an independent data set. When the aim is to perform prediction, it is used to evaluate how a predictive model will perform in general. In any problem, a model is usually given a dataset of known data on which training is run (training dataset) and a dataset of unknown data on which the model is tested (test dataset). It consists of dividing the dataset into complementary subsets, performing the analysis of training subset and validating the analysis of test data set. The various types of cross validation techniques are:

2.7.2.1 Leave-one-out cross-validation (LOOCV)

It is a special case of K-fold cross validation technique in which the whole dataset is divided into N number of datasets where N = K. Each time all the N-1 datasets are used as training datasets while leaving 1 different data set for the prediction. The error rate is computed by averaging the error rates from all the test datasets [16] [36] [44] [53] [54] [74] [77] [102] [124] [129] [131] [145] [148] [150].

2.7.2.2 K-Fold cross validation

In this type of cross validation, the data set is divided into K equal sized parts. The function approximator fits the values for K-1 parts of the data set and predicts the output value of the Kth part. Every time all the K-1 parts are treated as a training data set and a different Kth part is treated as test data set. The average error rate of testing dataset is calculated from all the iterations of Kth part. The general choices of K are 3, 5 and 10 [12] [16] [56] [57] [59] [69] [96] [110] [115] [122] [128] [134] [137] [139] [145] [148] [159].

Chapter 3

Diagnosis of Facioscapulohumeral Muscular Dystrophy using Cosine Distance Metric based Hierarchical Clustering and K Nearest Neighbor

In this chapter, a gene clustering model is designed using an unsupervised approach of feature selection to overcome the nasty issue of dimensionality reduction, thus leading to an accurate classification of NMD dataset. The designed model emphasizes the accurate classification by clustering the significant genes using the cosine distance metric based hierarchical clustering method. The whole chapter is organized in the following sections: Section 3.1 presents the introduction. Section 3.2 gives the methodology for gene selection and classification of FSHD. Section 3.3 presents the experimental results and their comparison. Section 3.4 concludes the chapter.

3.1 Introduction

The genetic diagnosis of NMD is an active area of research. The microarrays are used to examine the changes in the activity or expression level of genes for the accurate diagnosis. As the problem discussed in chapter 2, the dimension of the gene expression matrix is very large as compared to the number of publicly available samples. Hence, the dimensionality of the gene expression matrix needs to be reduced for the correct diagnosis. So, in the present chapter, we have made an intelligent integrated model for reducing the gene expression matrix dimension which further leads to the accurate classification of NMD datasets. An unsupervised approach of feature selection is implemented by the means of clustering methods.

The NMD chosen for the evaluation of the proposed integrated model is facioscapulohumeral muscular dystrophy (FSHD). It is an inherited, autosomal dominant NMD [161] [162]. The term ‘facioscapulohumeral’ is comprised of the name of muscles of the body, i.e., face (face), scapula (shoulder) and humeral (upper arm). This disease generally affects the upper body and causes the weakness in the muscles. It is caused due to the narrowing of polymorphic macrosattelite repeat D4Z4 on chromosome 4q35 [162]. According to the FSHD Global Research Foundation, a new estimate says that FSHD affects 1 out of the 7500 people. The symptoms of the disease are more prone in men than women.

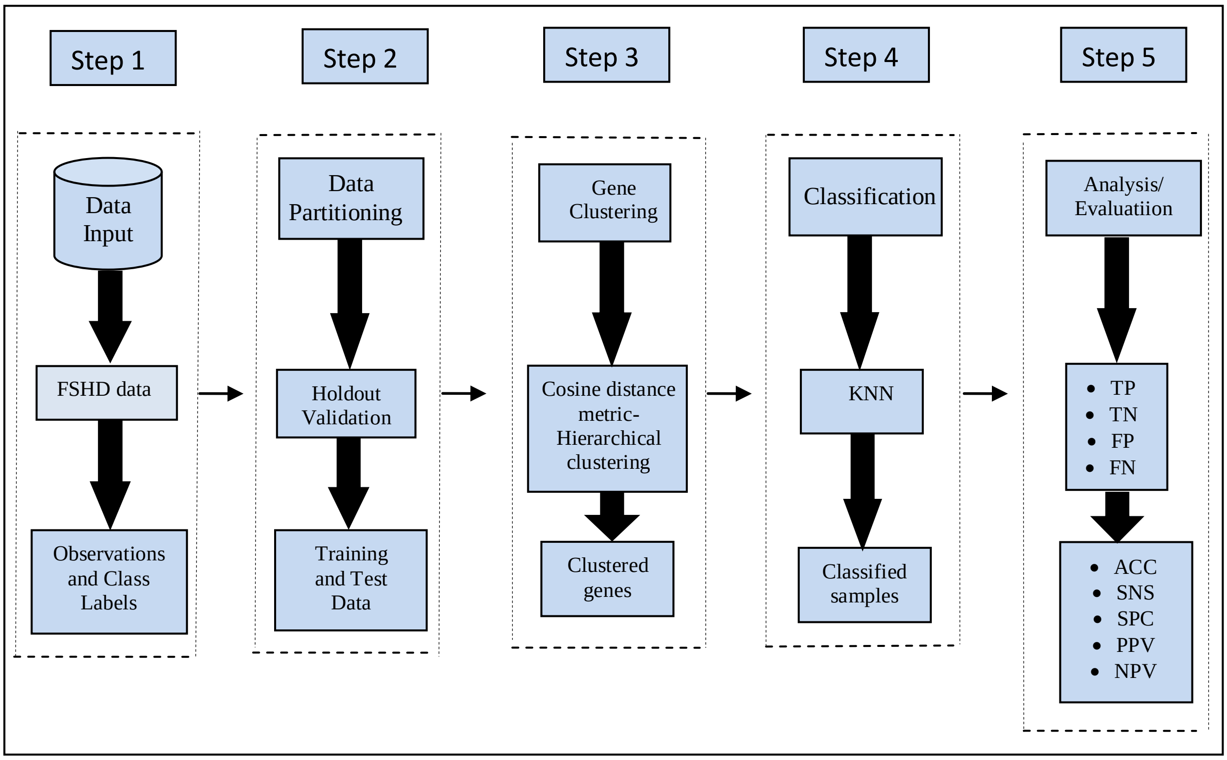
For the diagnosis of any NMD, first a neurologist sees the pattern of the muscles, which is usually done using electromyography (EMG). EMG depicts that the person is suffering from the disease, but not able to differentiate its kind. A physician always prefers the genetic testing for the diagnosis of any NMD. This is done by monitoring the gene expression levels using the microarray technology. In the genetic diagnosis of FSHD using microarrays, usually the blood samples are monitored. The smallest genetic changes are detected using the microarray technology by considering those blood samples.

Because of the typical nature of the FSHD microarray data sets, i.e., tens of thousands of genes, but a very few number of samples; which simply directs to the issue of overfitting. This typical nature of the data sets, poses a challenge in the correct diagnosis of the disease. The informative genes existing in the data set that are particularly related to the disease are very less. As a very less number of genes have different level of activity under the condition of interest, whereas a large numbers of the genes exhibit a similar expression profile, so they are not relevant to the classification task. Hence, to correctly diagnose the disease, there is a need to reduce the number of genes. This challenge is similar to the problem of feature selection in computer science where the main aim is to increase the classification accuracy. As stated earlier by employing the data reduction methods, we can increase the performance of classifiers by decreasing its computational burden. In addition to that we can reduce the training time, execution time, cost of classification and the risk of overfitting which will help in the efficient classification of the data set. So, an intelligent integrated model is proposed which first reduces the dimension of gene expression matrix and then does the accurate classification of the data set.

3.2 Methodology

The block diagram of the proposed model is shown in figure 3.1. In this chapter, we have implemented an unsupervised approach of gene selection, i.e., two types of clustering methods, namely, k-means and hierarchical clustering algorithm (with euclidean and cosine distance metrics) are deployed to cluster the informative genes. Before implementing clustering methods, Wilcoxon signed rank test is employed to rank the genes. To the ranked genes, clustering is applied. Followed by the clustering algorithms, three classification algorithms, LDA, QDA and KNN are implemented one at a time. So, the nine intelligent integrated approaches are implemented on the FSHD dataset DST-1 are: K-means-LDA, K-means-QDA, K-means-KNN, euclidean distance metric hierarchical clustering-LDA, euclidean distance metric hierarchical clustering-QDA, euclidean distance metric hierarchical clustering-KNN, cosine distance metric hierarchical clustering-LDA, cosine distance metric hierarchical clustering-QDA and cosine distance metric hierarchical clustering-KNN.

Figure 3.1: Block diagram of the proposed model



Clustering is the procedure of combining a number of data objects into multiple groups, subsets or clusters. While the data objects in a cluster are alike to each other and different to data objects in other clusters. That is why; sometimes it is called an automatic classification. The main component in a clustering algorithm is the distance between the attribute values which is used to assess the similarity and dissimilarity between the clusters. It is useful to discover the previous unknown groups within the data. Clustering is a form of learning from observations.

The clustering methods can be divided into two types: partition based clustering and connectivity based clustering. In this step, we have implemented both types of clustering methods on genes and the results are compared. From the former type, k-means clustering method is employed and from the latter type, hierarchical clustering is employed. Assume we have a set of *n* objects; the partitioning based clustering methods partitions the objects into *k* clusters. Here each cluster is represented by a partition with the condition k≤ n. The condition for making cluster is that the each cluster must contain at least one object. These clustering methods follow the simple rule of "restricted cluster separation" which means one object just belongs to one cluster. One type of partition based clustering method is k-means clustering algorithm. In the k-means clustering, the number of clusters needs to be prior defined. The criterion for making k clusters in k-means clustering algorithm is that the intra-cluster objects are more similar and inter-cluster objects are more dissimilar. The working of k-means algorithm is as follows: In the first step, *k* objects out of objects in *D* are randomly selected, initially these *k* objects represent the center or cluster head or cluster mean. For the outstanding objects in *D*, find out the most similar cluster for each object and the similarity is found based on the "Euclidean" distance between the cluster mean and the object. It keeps on iterating until it improves the similarity within the cluster. In every iteration, for every cluster it calculates the new mean of the objects in a cluster. The objects are reassigned according to the new mean of the clusters. The algorithm will stop iterating when the clusters formed in previous iteration is same as the clusters formed in the current iteration. The algorithm for k-means is given below.

Algorithm: K-means. For partitioning, while every cluster's head is presented by the mean value of objects in the cluster.

Input:

- k: number of clusters
- B: n objects in a data set.

Output: Data set is divided into the set of k clusters

Method:

(1) The initial cluster centers are chosen randomly by choosing k objects from D;

(2) Repeat

(3) (Re) assignment of each object to the latest cluster to which the object is most alike, based on the mean value of the objects in the cluster;

(4) Revise the cluster means.

(5) Until no change;

As we have seen that K-means clustering algorithm is based on centroid and develops in an iterative manner. Its main aim is to separate the *n* number of observations into *k* number of clusters, in which each observation lies in the cluster which is having the nearest mean. The clusters are made in such a way that the intra cluster distance is less and the inter-cluster distance is more. The centroid value of the cluster is adjusted in an iterative fashion every time, it means when a new gene is introduced, the distance between genes in intra cluster is less. The distance measure employed to measure the distance is "Squared Euclidean Distance". The default method "sample" is employed to pick the seeds, i.e., the initial cluster centroid position.

The hierarchical clustering method arranges the data objects in a tree or hierarchy fashion. It builds the tree of clusters in a hierarchy according to the distance metric which is called a dendrogram. In dendrogram, the entire data set is represented using the root node, whereas the leaf node is referred as a sample. There are two types of hierarchical method based on the procedure that how the hierarchical decomposition is created: agglomerative and divisive. In an agglomerative approach, it follows the bottom-up procedure. It initiates by creating a different cluster of each object. Then it starts merging the clusters into clusters and keeps on doing until it forms one cluster or a termination stage comes. That one cluster will be considered as the hierarchy's root. Two clusters which are found to be closer to each other are merged into one cluster. The divisive approach follows the top-down approach which initiates with one cluster having all the objects in it. In each iteration, it keeps on splitting the cluster in different smaller clusters until each object is only in one cluster. In both of the approaches, it is very difficult to find out the merge or split point. Here, the agglomerative approach is employed. A metric for distance function and a linkage criterion to specify the dissimilarity of sets is chosen. We have employed "euclidean" and "cosine" distance metrics for computing the distance between the objects and linkage clustering criterion used is "Unweighted average distance". The cluster tree formed using agglomerative clustering method is consistent as the cophenetic correlation coefficient given by the combination of euclidean distance metric and average linkage criterion is 0.7356 and the cosine distance metric and average linkage criterion is 0.9797. This extremely high value shows that the tree seems to be reasonably good fit to the distances. The challenge with both approaches of hierarchical clustering methods, we cannot undo anything or we cannot change anything of the previous step. The distance between two clusters is computed. The various distance metrics can be euclidean distance or cosine distance which is shown in equations 3.1 and 3.2 respectively.

$$d_{st}^2 = (x_s - x_t)(x_s - x_t)' \tag{3.1}$$

$$d_{st} = 1 - \frac{x_s x_t'}{\sqrt{(x_s x_s')(x_t x_t')}} \tag{3.2}$$

The genes are clustered from the training data set only. Our goal is to identify the class of new samples (in the test data) based on the training data. It is considered as only a two-class classification problem as defined in chapter 2, as our classification task is to identify whether the person has FSHD (diseased) or not (healthy). For this experimentation task, three efficient classifiers are selected, namely, LDA, QDA and KNN which classifies the samples after clustering the genes. The detailed description of these classification algorithms is given in chapter 2.

3.3 Results and Discussion

Initially, we run holdout validation technique to assess the performance of the model made for clustering of genes and thus for the classification of samples. This method is very fast, easy and is considered to give unbiased results. Its working is explained in section 2.7. The data set used for the evaluation of the proposed technique is DST-1 which is detailed in section 2.3. For the data set DST-1, the performance of two clustering algorithms with different parameters and various classifiers are shown in tables 3.1, 3.2 and 3.3. Various performance measures listed in section 2.5 are calculated for the nine implemented intelligent integrated methods. Table 3.1 indicates the performance measures of using k-means clustering method with LDA, QDA and KNN. Table 3.2 represents the performance measures of using euclidean distance metric hierarchical clustering method with LDA, QDA and KNN. Table 3.3 shows the performance measures of using a cosine distance metric hierarchical clustering method with LDA, QDA and KNN. The linkage criteria employed in both cases of hierarchical clustering method are “average linkage criterion”.

For the k-means clustering algorithm, the value of k is chosen to be 25. The clusters are made with k-means clustering method and the samples are first classified using LDA.

Here, the training data set shows the performance measures as ACC (59.68%), SNS (58.43%), SPC (60.82%), PPV (57.78%) and NPV (61.46%) whereas the results on test data set are ACC (61%), SNS (60.42%), SPC (61.54%), PPV (59.18%) and NPV (62.75%). The performance measures of this integrated method are found to be unsatisfactory. So, the next classification algorithm employed is QDA, i.e., k-means-QDA for the clustered genes. The results of this integration were also not up to the mark as the performance on the training data set is ACC (60.42%), SNS (55.28%), SPS (65.14%), PPV (59.33%) and NPV (61.29%). The result of the test data set is ACC (60.86%), SNS (55.95%), SPS (65.38%), PPV (59.87%) and NPV (61.66%). So for the further improvement, the KNN is employed for the classification of the samples. The training data set leaves us with ACC (65.71%), SNS (56.25%), SPC (73.68%), PPV (64.29%), and NPV (66.67%). The performance of test data set is ACC (62%), SNS (45.83%), SPC (76.92%), PPV (64.71%) and NPV (60.61%). Table 3.1 shows all the results of using k-means clustering and three different classifiers LDA, QDA and KNN.

Table 3.1**:** Performance measures of k-means clustering algorithm with LDA, QDA and KNN.

Integrated method	**Data set**	**ACC**	**SNS**	**SPC**	**PPV**	**NPV**
K-means-LDA	Training	59.68%	58.43%	60.82%	57.78%	61.46%
	Test	61%	60.42%	61.54%	59.18%	62.75%
K-means-QDA	Training	60.42%	55.28%	65.14%	59.33%	61.29%
	Test	60.86%	55.95%	65.38%	59.87%	61.66%
K-means-KNN	Training	65.71%	56.25%	73.68%	64.29%	66.67%
	Test	62%	45.83%	76.92%	64.71%	60.61%

After that, another clustering algorithm, i.e., hierarchical clustering algorithm is employed. The distance metrics used for measuring the distance of the observations are “cosine” and “euclidean”. First euclidean distance metric based hierarchical clustering method with three classification algorithms are employed, from that it was found that the results are not acceptable for these integrations too. When euclidean distance metric hierarchical clustering algorithm is employed with LDA, the performance of the training data set is ACC (50%), SNS (52.94%), SPC (47.37%), PPV (47.37%) and NPV (52.94%) and of the test data set is ACC (54%), SNS (58.33%), SPC (50%), PPV (51.85%) and NPV (56.52%). Then in the same order as mentioned above QDA is used; this integration leaves us with poor results as on the training data set, it is ACC (52.33%), SNS (56.10%), SPC (48.89%), PPV (50%) and NPV

(55%). The results of the test data set are ACC (54%), SNS (58.33%), SPC (50%), PPV (51.85%) and NPV (56.52%). A little better results are found when KNN is integrated with the euclidean hierarchical clustering algorithm as in the training data set and test data set are ACC (56%), SNS (53.85%), SPC (57.75%), PPV (53.85%), NPV (57.75%) and ACC (56%), SNS (52.78%), SPC (58.97%), PPV (54.29%) and NPV (57.50%) respectively. The summarization of the results of this integration is shown below in table 3.2.

Table 3.2: Performance measures of euclidean distance metric based hierarchical clustering algorithm with LDA, QDA and KNN.

Integrated method	**Data set**	**ACC**	**SNS**	**SPC**	**PPV**	**NPV**
Euclidean distance metric based hierarchical clustering – LDA	Training	50%	52.94%	47.37%	47.37%	52.94%
	Test	54%	58.33%	50%	51.85%	56.52%
Euclidean distance metric based hierarchical clustering – QDA	Training	52.33%	56.10%	48.89%	50.00%	55.00%
	Test	54%	58.33%	50%	51.85%	56.52%
Euclidean distance metric based hierarchical clustering – KNN	Training	56%	53.85%	57.75%	53.85%	57.75%
	Test	56%	52.78%	58.97%	54.29%	57.50%

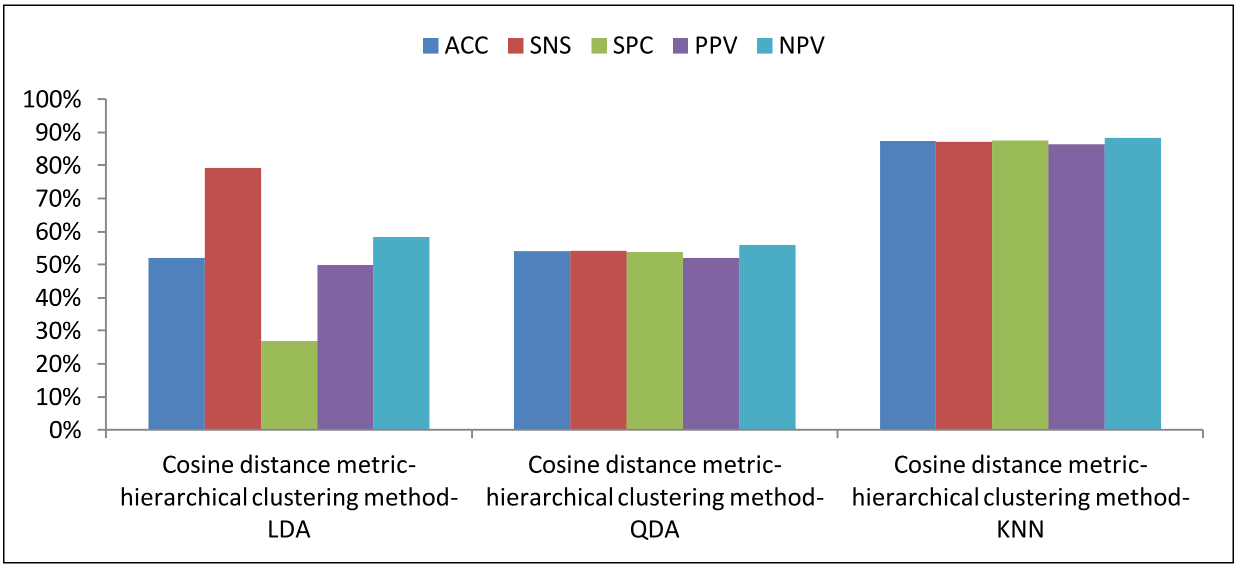
In another case of hierarchical clustering algorithms, i.e., the use of cosine distance metric with average linkage criterion, the results given by KNN is the most superior, whereas the performance of LDA and QDA is found to be unsatisfactory. When LDA is used, the results of the training data set and test data set is ACC (47.22%), SNS (70.59%), SPC (26.32%), PPV (46.15%), NPV (50%) and ACC (52%), SNS (79.17%), SPC (26.92%), PPV (50%) and NPV (58.33%) respectively. The lowest specificity is given by this integrated method. Next, the QDA is integrated with the cosine distance metric based hierarchical clustering algorithm, the outcome of this on the training data set and test data set is ACC (54.65%), SNS (58.54%), SPC (51.11%), PPV (52.17%), NPV (57.50%) and ACC (54%), SNS (54.17%), SPC (53.85%), PPV (52%) and NPV (56%) correspondingly. The best results are given by the integration of cosine distance metric based hierarchical clustering algorithm with KNN as on the training data set; the performance is ACC (87.93%), SNS (87.68%), SPC (88.15%), PPV (87%), and NPV (88.78%) and on a test data set is ACC (87.39%), SNS

(87.19%), SPC (87.57%), PPV (86.39%), and NPV (88.39%). The result of this integration is presented in table 3.3. The difference in performance of cosine distance metric based hierarchical clustering algorithm with LDA, QDA and KNN is shown in figure 3.2.

Table 3.3: Performance measures of cosine distance metric based hierarchical clustering algorithm with LDA, QDA and KNN.

Integrated method	**Data set**	**ACC**	**SNS**	**SPC**	**PPV**	**NPV**
Cosine distance metric based hierarchical clustering – LDA	Training	47.22%	70.59%	26.32%	46.15%	50%
	Test	52%	79.17%	26.92%	50%	58.33%
Cosine distance metric based hierarchical clustering – QDA	Training	54.65%	58.54%	51.11%	52.17%	57.50%
	Test	54%	54.17%	53.85%	52%	56%
Cosine distance metric based hierarchical clustering – KNN	Training	87.93%	87.68%	88.15%	87.00%	88.78%
	Test	87.39%	87.19%	87.57%	86.39%	88.31%

Figure 3.2: Graphical illustrations of performance measures of intelligent integrated method cosine distance metric based hierarchical clustering method – three classifiers



The most challenging task in employing k-means clustering algorithm is the value of k, i.e., the number of clusters necessitates being priory defined. We have tried different values of k in the k-means, the optimum results were given by K=25 only. K-means algorithm also shows the problem when the size of clusters are different. The results of k-means algorithm are also prone to local minima. A hierarchical clustering algorithm is always preferred because it is not necessitate to specify the number of clusters and it usually gives higher performance [163]. As we can see that the cophenetic correlation coefficient using cosine distance metric is found to be better as compared to the euclidean distance metric. The hierarchical clustering algorithm with cosine distance metric is giving high performance because it considers the relative sizes rather than the absolute sizes of observations.

3.4 Conclusions

The process of clustering the genes in a high dimensional dataset is a very crucial step for the diagnosis of a disease. In this chapter, two clustering methods are employed, namely k-means and hierarchical clustering algorithm for clustering the genes and LDA, QDA and KNN are deployed for the classification task for the genetic diagnosis of an NMD, i.e., FSHD. The results of the various integrations indicate that the best performance is given by cosine distance metric based hierarchical clustering algorithm-KNN. The proposed integrated model effectively clusters the genes, this shows that the data clustering with the proper parameters for such a high dimensional data play a vital role in achieving good classification results. Hence, these unsupervised methods can be used for the clustering the genes which can be further used for the classification of samples of other NMDs.

Though this integrated technique has given high performance measures, still it is difficult to assess the significance of the features. This problem can be resolved with the use of supervised methods of feature selection techniques. Because supervised feature selection techniques assesses the relevance of features guided by the label information. While the unsupervised feature selection works with unlabeled data, so it is very difficult to assess the significance of features.

The work presented in this chapter has been published in International Journal of E-Health and Medical Communications, Vol. 8, No. 2, pp. 33-46, 2017, IGI Global Publishing, United States, DOI: 10.4018/IJEHMC.2017040103, mentioned under the list of publications at the end of chapter 8.

Chapter 4

An Intelligent Integrated Method for Dimension Reduction and Classification Applied to the Microarray Data of Neuromuscular Dystrophies

As observed in the previous chapter, the unsupervised feature selection methods have limitation in accessing the significance of data. This problem can be resolved by using the supervised feature selection methods. Therefore, in this chapter, a gene selection model is designed using the supervised approach of feature selection to overcome the issue of dimensionality reduction, and thus leading to the accurate classification of NMDs. Here, the filter models of feature selection are employed to select the significant genes. This chapter is structured as follows: Section 4.1 presents the introduction. Section 4.2 gives the methodology for gene selection and classification of NMDs. Section 4.3 presents the experimental results and their comparison. Section 4.4 concludes the chapter.

4.1 Introduction

The microarray technology allows the NMD to be predicted using gene expression activity, i.e., it is used to monitor the whole genome of a given organism simultaneously. They provide us a path to obtain the expression level of a large number of genes at a single time under a particular condition in an experiment [164]. In microarrays, gene expression matrix is formed where samples are represented in columns and genes are represented in rows. Each cell in a gene expression matrix signifies the expression value of a gene in a sample. These days, the microarrays are used to diagnose the diseases using these gene expression values [47]. As also stated earlier, the classification of NMDs using the microarray data is a bit complex task because of the following reasons. Firstly, the microarray data is high dimensional as it contains a huge number of genes, i.e., tens of thousands of genes. Secondly, it contains very less samples, i.e., a very few number of patients. Thirdly, from all these genes, only a few genes are related to the diseases, rest other genes are noisy [165].

So, we need to decrease the number of genes in order to properly diagnosis the disease and to get the accurate diagnosis results. In this chapter, we have implemented two supervised methods for gene selection as the unsupervised methods implemented in the last chapter did not give the satisfactory results [166]. As also stated earlier, the supervised feature selection methods are considered better because they assess the relevance of features guided by the label information. While the unsupervised feature selection methods work with unlabeled data, so it is very complicated to evaluate the significance of features. The supervised feature selection methods can be further categorized into three types depending upon how the feature selection search is combined with the classification model: filter models, wrapper models and embedded models as discussed in section 2.4. From the above three categories, in this chapter we have implemented the filter models to reduce the gene expression data dimension. Two filter models, namely t-test and entropy; and two classification algorithms, i.e., KNN and linear SVM are employed for the feature selection and classification of NMDs data sets.

Two NMDs Juvenile Dermatomyositis (JDM) and FSHD are used for employing the integration of these techniques. JDM is an autoimmune disease which generally affects children. It usually affects 3000-5000 children in United States each year. It is a genetic disease which tends to run in the families of patients. In this, the muscle weakness may results in dysphonia, fatigue, weight loss, clumsiness and other issues. FSHD is also an autosomal dominant NMD. The muscle weakness mostly occurs in the muscles of face, shoulder, arms and stomach. As such, there is no cure for these diseases. But a proper and accurate diagnosis of the disease could help the patient in many ways. Both the data sets are very dimensional and their detailed description is given in sections 2.3.

4.2 Methodology

Classification is the process of identification of categories of test observations, i.e., unseen data on the basis of the training observations, i.e., seen data. Feature selection for classification selects the subset of highly discriminating features from the training data. The overall framework of the integrated model is illustrated in figure 4.1. Two supervised filter gene selection methods, namely t-test and entropy are deployed on the training data set. Two classification methods KNN and linear SVM are implemented to classify the NMD data sets after the selection of genes. The integrated methods employed are t-test-KNN, t-test-linear SVM, entropy-KNN and entropy-linear SVM. The performance of these integrated algorithms is evaluated using various performance measures as discussed in chapter 2.

For ranking and selection of genes, t-test is applied and the p-value is calculated. The procedure for calculating p-value using t-test is:

1) Compute the mean (M) of the gene expression value of the diseased samples for the first gene.
2) Compute the M of gene expression value of the non-diseased or normal samples for the first gene.
3) Calculate the difference between the average of diseased samples and non-diseased samples.
4) Calculate the standard deviation (S.D.) of gene expression value of the diseased samples for the first gene.
5) Calculate the S.D. of gene expression value of the non-diseased or normal samples for the first gene.
6) Use the standard deviation to calculate the p-value for the first gene.
7) Calculate for all the remaining genes in the samples. The p-value is calculated using the formula given in equation 4.1

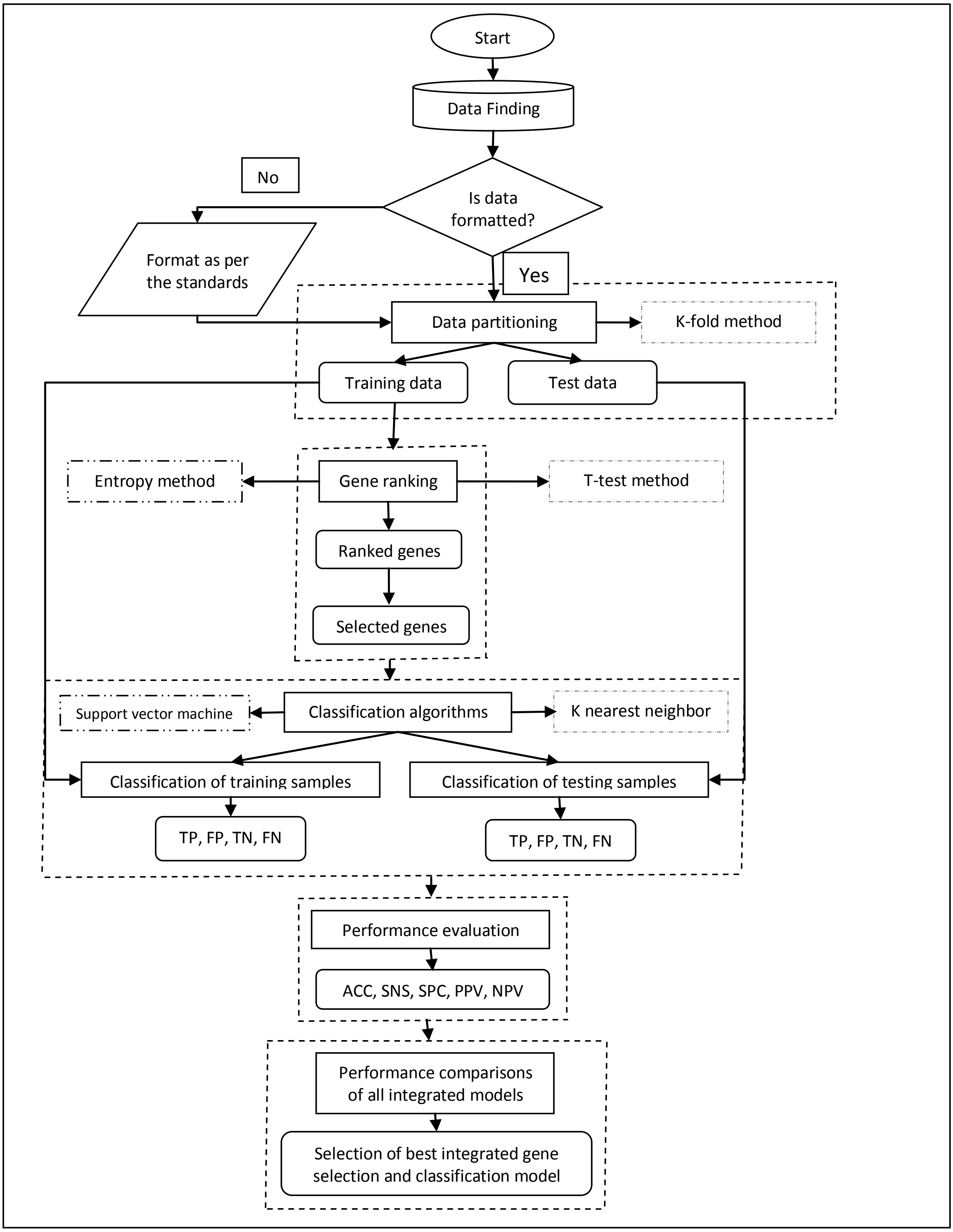
$$p - value = max\left\{\left|\frac{\bar{p}_{ij} - \bar{p}_i}{m_{jk}s_i}\right|, j = 1,2,3, \dots \dots \dots \dots J\right. \qquad (4.1)$$

where p_{ij} refers to the gene expression value of gene i in sample j, $\bar{p}_{ij}$ is the mean gene expression value of gene i in sample j, $\bar{p}_i$ is the general gene expression value of gene i, s_i refers to the within class S.D. of gene i. The p-value of all the genes is computed and all the genes are rearranged according to their p-value in descending order. Some of the top genes are chosen and given as an input to the classification algorithms.

Another gene selection method employed is entropy. It is also known as information divergence or Kullback-Leibler divergence or relative entropy. It is employed to quantify the divergence between two probability distributions [167]. Suppose the probability functions of two discrete distributions A and B are A_k and B_k respectively. Then the distance of A with respect to B is given in equation 4.2.

$$e = \sum_k A_k \log_2\left(\frac{A_k}{B_k}\right) \qquad (4.2)$$

Figure 4.1: Overall framework



The relative entropies of all the genes are calculated. The genes are arranged in the descending order of their value. From them, some top genes are selected and are further used for classification using KNN and linear SVM. The detailed description of KNN and linear SVM are given in subsections 2.5.3 and 2.5.4 respectively.

4.3 Results and Discussion

The proposed integrated methods are evaluated on the data sets, i.e., DST-2 and DST-3 given in subsections 2.3.2 and 2.3.3 respectively. We conduct fivefold cross validation experiments on the data sets. The description of fivefold cross validation is given in section 2.7. The performance measures of various integrations which are implemented like t-test–linear SVM, t-test–KNN, entropy–linear SVM and entropy–KNN on DST-2 and DST-3 data sets are calculated as shown in section 2.6 and compared. Table 4.1 and figure 4.2 illustrate the results of the integrations t-test–linear SVM and t-test–KNN on DST-2 data set. Table 4.2 and figure 4.3 shows the results of integrations entropy-linear SVM and entropy-KNN on DST-2 data set. Table 4.3 and figure 4.4 depict the results of using t-test–linear SVM and t-test–KNN on DST-3 data set. Table 4.4 and figure 4.5 represent the results of integrations entropy-linear SVM and entropy-KNN on DST-3 data set.

Table 4.1: Performance measures of integration of t-test–linear SVM and t-test–KNN on DST-2

Integrated method	Data	Accuracy	Sensitivity	Specificity	PPV	NPV
T-test-Linear SVM	Training	78.87%	78.95%	78.79%	81.80%	76.47%
	Test	74.36%	76.19%	72.22%	76.19%	72.22%
T-test-KNN	Training	100%	100%	100%	100%	100%
	Test	89.74%	90.48%	88.89%	90.48%	88.89%

In case of other integrated algorithms, namely entropy–linear SVM and entropy–KNN, the classifiers have shown different performance after ranking and selecting the genes using entropy. The classifier KNN has shown the superior performance than linear SVM. It has given 100% performance measures in the training data set and better accuracy, i.e., 92.31% in the test data set as compared to the previous integrated algorithm.

Figure 4.2: Integration of t-test–linear SVM and t-test–KNN on DST-2

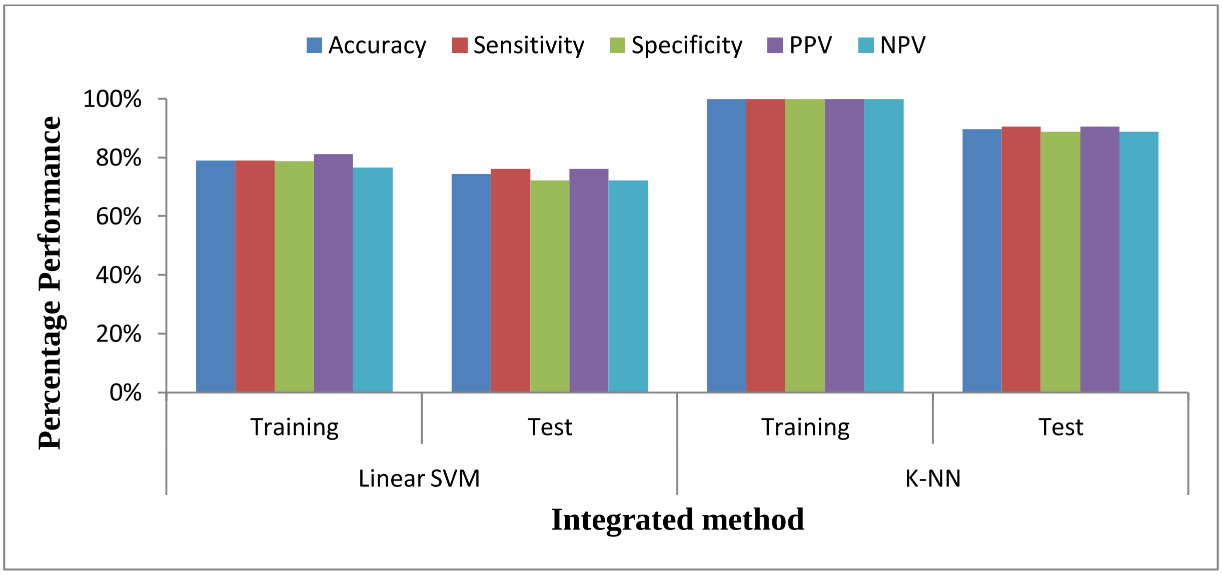


Table 4.2: Performance measures of integration of entropy–linear SVM and entropy–KNN on DST-2

Integrated method	Data	Accuracy	Sensitivity	Specificity	PPV	NPV
Entropy-Linear SVM	Training	76.06%	86.84%	63.64%	73.33%	80.77%
	Test	74.36%	88.10%	58.33%	71.15%	80.77%
Entropy-KNN	Training	100%	100%	100%	100%	100%
	Test	92.31%	90.48%	94.44%	95%	89.47%

Figure 4.3: Integration of entropy–linear SVM and entropy–KNN on DST-2

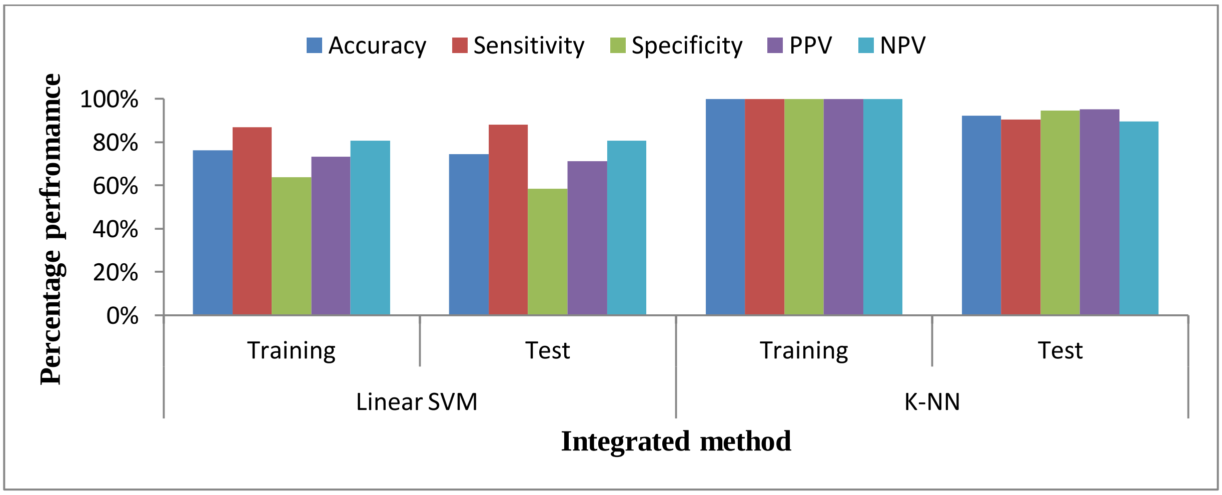
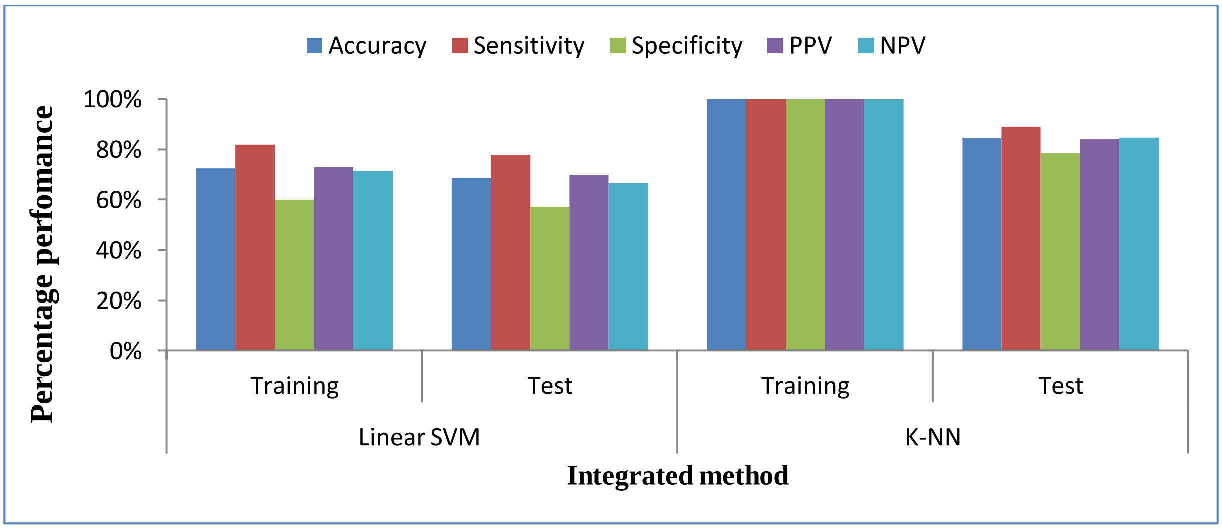


Table 4.3: Performance measures of integration of t-test–linear SVM and t-test–KNN on DST-3

Integrated Method	**Data**	**Accuracy**	**Sensitivity**	**Specificity**	**PPV**	**NPV**
T-test-Linear SVM	Training	72.41%	81.82%	60%	72.97%	71.43%
	Test	68.75%	77.78%	57.14%	70%	66.67%
T-test-KNN	Training	100%	100%	100%	100%	100%
	Test	84.38%	88.89%	78.57%	84.21%	84.62%

Figure 4.4: Integration of t-test–linear SVM and t-test–KNN on DST-3



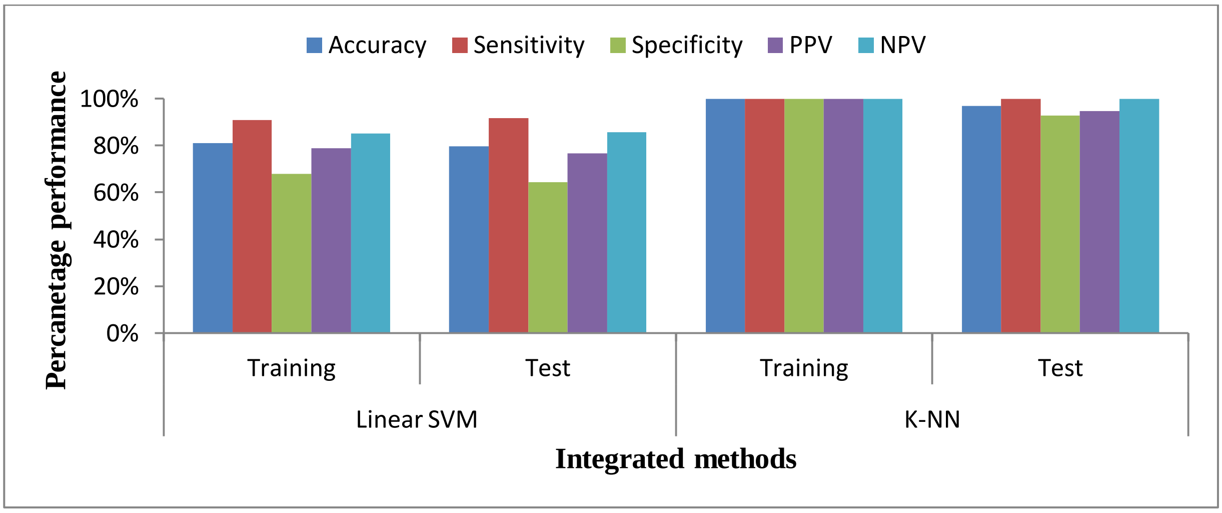
Here also, the integrated algorithm t-test-KNN has outperformed t-test–linear SVM. The training data set has given 100% of all the performance measures, whereas the accuracy of test data set is 84.38%. The best results on both datasets were found when the entropy is integrated with KNN. Like the previous integration, the training data set has given 100% of all the performance measures and test data set has given 96.88% accuracy.

Table 4.4: Performance measures of integration of entropy–linear SVM and entropy–KNN on DST-3

Integrated method	**Data**	**Accuracy**	**Sensitivity**	**Specificity**	**PPV**	**NPV**
Entropy-	Training	81.03%	90.91%	68%	78.95%	85%

Linear SVM	Test	79.69%	91.67%	64.29%	76.74%	85.71%
Entropy-KNN	Training	100%	100%	100%	100%	100%
	Test	96.88%	100%	92.86%	94.74%	100%

Figure 4.5: Integration of entropy–linear SVM and entropy–KNN on DST-3



We observe that the classifiers do not result same after ranking and selecting the genes using t-test. The integration t-test–KNN is found to be more efficient as it is giving 100% accuracy in the training data set and 89.74% accuracy in the test data set.

4.4 Conclusions

In case of NMDs, it is a complex task to find disease specific genes which assists in accurately diagnosing the diseases. In the present chapter, the proposed integrated algorithm selects genes and classifies the microarray data of FSHD and JDM. The genes are ranked and selected using t-test and entropy based supervised gene selection methods, which can successfully classify the samples using KNN and linear SVM. The integration of entropy with KNN has given the best performance measures for gene selection and classification of both the data sets and it can also be applied for the diagnosis of other NMDs using microarray data.

Despite the fact that filter models of supervised feature selection techniques have given the better performance than the unsupervised techniques, but still it ignores the feature independencies and the interaction with the classifiers. This problem can be resolved by using the embedded models of the feature selection techniques because they interact with the classifiers and improves the value of performance measures.

The work presented in this chapter has been published in Indian Journal of Science and Technology, Vol. 9, No. 28, pp. 1-6, 2016, India, DOI: 10.17485/ijst/2016/v9i28/98378, mentioned under the list of publications at the end of chapter 8.

Chapter 5

Building an Intelligent Integrated Method of Gene Selection for Facioscapulohumeral Muscular Dystrophy Diagnosis

In this chapter, a gene selection model is designed by integrating the filter and embedded models of supervised feature selection. From the filter model, t-test is chosen and from the embedded model, GA is used for the optimization of features. The integration of t-test and GA leads to the accurate classification of NMD datasets. Here we are selecting the genes for the binary class classification problem. So, this chapter is structured as follows: Section 5.1 presents the introduction. Section 5.2 gives the methodology for gene selection and classification of NMDs. Section 5.3 presents the experimental results and their comparisons. Section 5.4 concludes the chapter.

5.1 Introduction

An NMD affects the neuromuscular system of a human body. According to the Muscular Dystrophy Association, in US more than a million people are affected by some kind of NMD. In the present chapter, we have chosen the same genetic NMD of the previous chapters, i.e., FSHD for the evaluation of the methods. This disease is formed due to the contraction of polymorphic macrosattelite repeat D4Z4 on chromosome 4q35 [162]. This disease mostly affects the muscles of face, shoulder and an upper arm as the name facio-scapulo-humeral signifies. It is caused due to the mutation in one or more genes. These diseases are very progressive in nature. Till date, we do not have any cure for most of the NMDs. Therefore, there is a need to diagnose these diseases in the early stages. These days, the genetic testing is considered as the most preferred way for the diagnosis of NMD [162]. The genetic testing is done by examining the expression levels of genes using microarrays. The microarray technology helps in detecting the smaller changes in DNA or chromosomes in human body [115].

Generally, the microarray data of FSHD are high dimensional as already discussed in chapter 2. The management and extraction of this large amount of data has become a challenge as it contains a huge number of genes but a very less number of samples. In a gene expression matrix, a sample is represented in the column and a gene is represented in row.

The expression values in these microarrays are represented through gene expression matrix. The gene expression value in a gene expression matrix is defined as $A = (A_{bc})$. The value A_{bc} represents the expression value of gene b at sample c. From these large numbers of genes, a very few genes are particularly related to the disease and rest other genes are uninformative. So, for the accurate diagnosis of these diseases, there is a need to diminish the number of uninformative genes. Hence, dimensionality reduction or choosing a smaller subset of genes is an important task in the diagnosis of such diseases. In diagnosing a disease using microarray data, the gene selection is usually performed. The gene selection methods reduce the computational complexity and increase the value of performance measures. These methods will help us to speed up the algorithm execution and to increase the prediction accuracy [128].

In the last chapter, we have implemented two filter models of feature selection, but the results provided by those methods were not up to the mark [168]. So, in the present chapter, the diagnosis of FSHD is done in three stages. The system design employed in the present chapter is illustrated in Figure 5.1. In the first stage, a filter model, namely t-test is implemented to preselect the genes by removing the most uninformative and redundant genes. In the second stage, GA is implemented as an embedded model. Here, three experiments are done on GA, wherein the fitness function is evaluated using LDA, QDA and KNN one after the other with varying number of genes to choose the most informative gene subset, the general idea of stage 2 is demonstrated in figure 5.2. In the third stage, the classification is done using the above mentioned classifiers. The experimental results of the implementation of these integrated algorithms on the FSHD dataset enable us to select a small subset of genes and get appreciable performance measures. The result shows that the integration of GA with KNN is found to be the best for gene selection and diagnosis of FSHD.

5.2 Methodology

The FSHD data set for the evaluation of these integrated methods is DST-1 which is taken from Gene Expression Omnibus (GEO) with id (GEO accession GSE36398) entitled “Transcriptional profiling in facioscapulohumeral muscular dystrophy to identify candidate biomarkers” [152]. The detailed description is given in section 2.3.1. The system design employed in the present chapter is demonstrated in figure 5.1. The task of feature selection is accomplished in two stages, i.e., stage 1 and stage 2. In stage 1, a filter model is implemented on the data set to remove out the most uninformative and redundant genes. In stage 2, GA is deployed wherein the fitness function is evaluated using three different classification algorithms, one at a time on the varying number of genes. In each experiment, the fitness function is evaluated using LDA, QDA and KNN is used in first, second and third experiment respectively which is shown in figure 5.2. After that in Stage 3, three classifiers mentioned above are employed to classify the FSHD data. The values of all performance measures of all three experiments with varying number of genes are given in table 5.2.

Stage 1: Preselection of genes

In stage 1, a filter technique, namely t-test is applied to preselect or to remove out the redundant and uninformative genes. This process evaluates each and every gene in the dataset and sorts them according to their relevance. From that, the candidate set is formed by selecting first p genes. Then these genes are passed on to the subsequent stage, i.e., the selection of genes. The procedure for calculating p-value using t-test is given in section 4.2.

Stage 2: Selection of genes

At this stage, to choose the highly informative and discriminatory genes from the candidate set generated from stage 1, GA is deployed as an embedded model. GA was first described by John Holland in 1960 [169]. GA belongs to the class of evolutionary computation which uses the computational model of evolutionary processes for solving any problem.

It is an iterative, non-traditional optimization technique and belongs to the category of EAs. They imitate the process of natural development and replicate the survival of the fittest among individuals over repeated generations for solving a problem because the most important goal of GA is continuous improvement in small steps. The main component in GA is a chromosome. The various genetic operators in GA are: reproduction, crossover and mutation. The GA parameters used in the present chapter are given in table 5.1.

Chromosome and initial population

The chromosomes are the candidate solution to the problem. In GA, each feature is encoded as a gene and can be symbolized in the form of 1s and 0s. The value 1 indicates the inclusion in the gene subset, else it will be excluded. A chromosome is defined as the set of genes. In the present chapter, the number of genes is varying as N=5, N=10, N=15, N=20 and

N=25. The population is referred as a collection of chromosomes. GA searches for a population of chromosomes not a single chromosome. The population size varies according to the nature of the problem and determines the number of chromosomes in each generation. The population size is chosen to be 100. The initial population of chromosomes is randomly generated. The number of genes was varied till 25 because it was found that there was just a slight difference between the values of performance measures and we have also found that the best accuracy was achieved when the number of genes was taken to be 10 only.

Fitness function

An objective function or fitness function improves the population, with each generation. The designing of fitness function is the most crucial task in using GA. In many cases, the fitness function is accessed by the classification performance of the classifier. In this chapter, firstly LOOCV is applied to calculate the average of performance measures. Its detailed description is given in section 2.7. We use the combination of error rate and conditional probability of the classifier as the fitness function calculated using LOOCV. Higher the value of the fitness function, more are chances to move to the next generation. This is shown in figure 5.2.

Reproduction (Selection)

To move to the next generation, selection is made from a portion of existing or current population, which becomes parent to the next generation. This is done on the basis of evaluation of fitness function as among all the parents, the fittest will survive and reproduce. The more the value of the fitness function, the more likely it is moving to the next generation. The various selection functions are roulette wheel selection, stochastic universal sampling, truncation selection and tournament selection. Here, we have chosen stochastic universal sampling because it uses evenly spaced intervals to sample all the solutions and to select a single random value [170].

Figure 5.1: System design

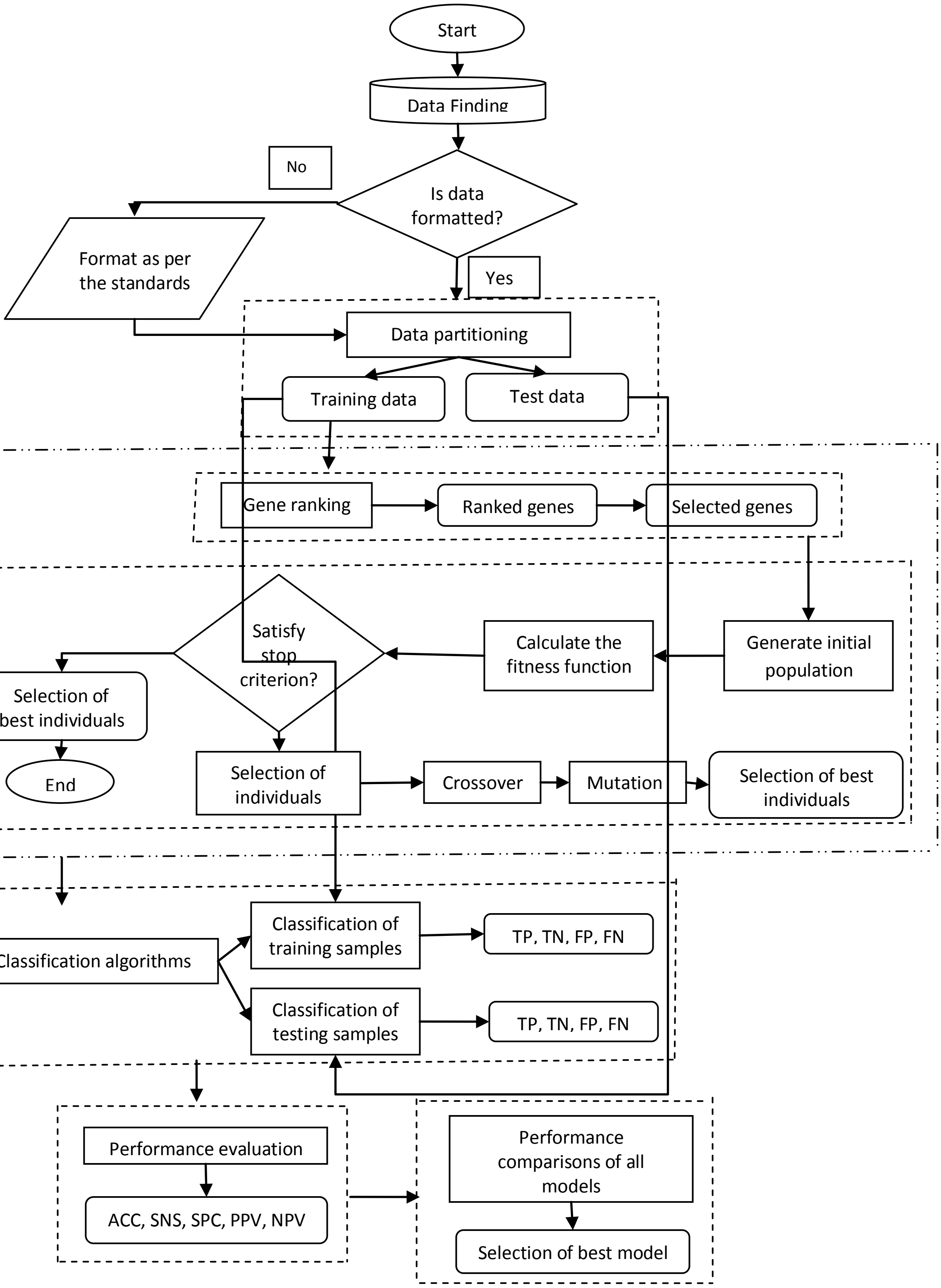
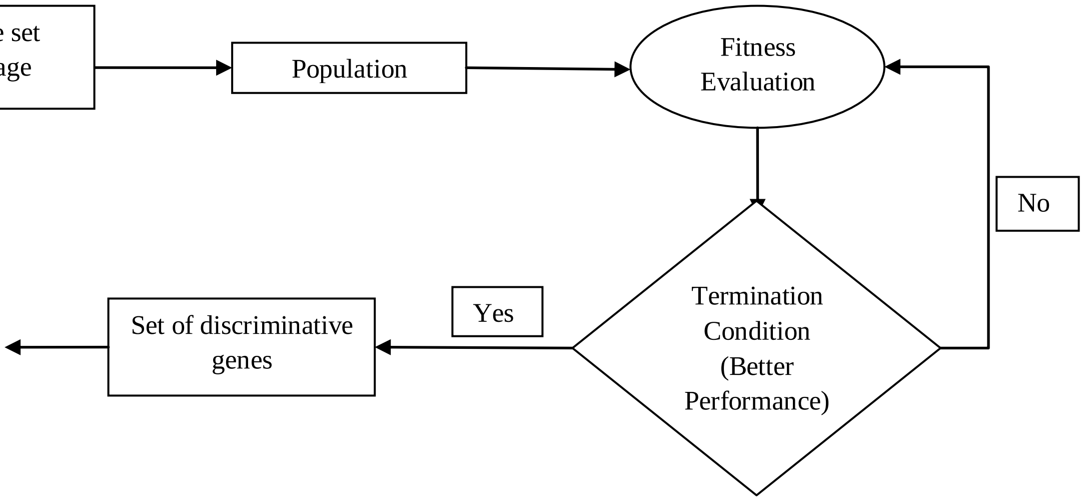


Figure 5.2: Structure of GA method


Candidate set from 1st stage
Population
Fitness Evaluation
No
Termination Condition (Better Performance)
Yes
Set of discriminative genes

Crossover

Crossover operation forms a new individual for the next generation by combining two parents as half of the chromosome from one parent and the other half of the chromosome of another parent are combined and forms a new child. The various types of crossover operators are scattered crossover, single-point crossover, two point crossover, intermediate crossover, heuristic crossover, arithmetic crossover, uniform crossover and flat crossover. In the present chapter, we have chosen scattered crossover function with a crossover probability of 0.7.

In some cases, it may happen that the new child may have sometimes the poor efficiency because of the repeated cutting, matching and mixing which results in a drop in quality. If this happens, elitist strategy can be used to overcome the situation in which the best solution in a particular generation is left untouched and the best solution automatically goes to the next generation.

Mutation

The diversification in the search space is preserved by the mutation operator. In mutation, gene value is changed from 0 to 1 and 1 to 0. The various types of mutation operator are gaussian mutation, bit string mutation, uniform mutation, non-uniform mutation, boundary mutation and flip bit mutation. In the present chapter, uniform mutation is used in which the chosen gene is replaced with the uniform random value. The mutation probability was taken to be 0.1.

Stopping criterion

The GA algorithm will work till a predefined number of iterations is reached or value of the objective function is reached to 100%. The stopping criterion here is 20 generations or till the classification performance has reached 100%.

Table 5.1: GA parameters

Parameter Name	**Value**
Chromosome	A bit string of 1s and 0s
Number of genes	5,10,15,20,25
Population Size	100
Selection Operator	Stochastic universal sampling
Fitness function	Calculated using LDA, QDA and KNN
Crossover Fraction	0.7
Mutation Operator	Uniform mutation
Mutation Rate	0.1
Stopping Criteria	20 generations

Stage 3: Classification

The subset of the most discriminatory genes is formed after stage 2. It is used to diagnose the disease which will classify whether the sample is diseased or not. The classification is done using LDA, QDA and KNN. The complete description of these classification algorithms is given in section 2.5. For KNN, the default value, i.e., K=1 is chosen with euclidean distance metric and nearest rule to classify the data.

5.3 Results and Discussion

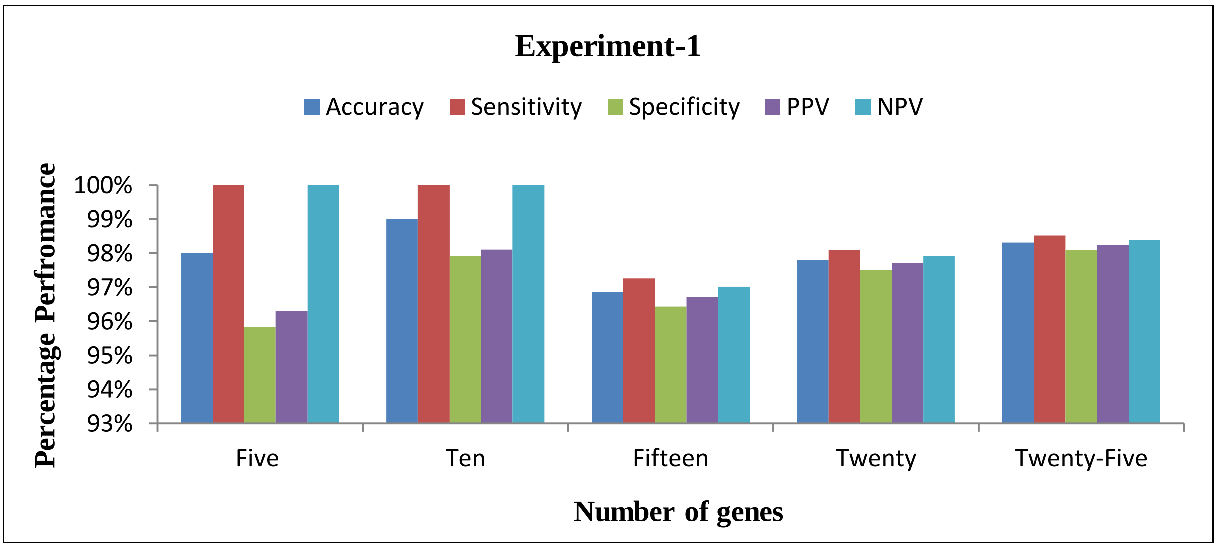
In this section, we present broad experiments performed on the FSHD dataset. Here, we can consider the diagnostic test as a binary classification problem in which the output can be either true or false. The method will be considered accurate if a diseased sample is diagnosed true and a healthy sample is diagnosed false. The performance measures of these experiments calculated are given in section 2.6.

Table 5.2**:** Performance measures

Experiment	**Number of genes selected**	**Accuracy (%)**	**Sensitivity (%)**	**Specificity (%)**	**PPV (%)**	**NPV (%)**
Experiment-1 (The fitness function is calculated using LDA)	5	98	100	95.83	96.30	100
	10	99	100	97.92	98.11	100
	15	96.86	97.25	96.43	96.72	97.01
	20	97.80	98.08	97.5	97.70	97.91
	25	98.31	98.52	98.08	98.23	98.39
Experiment-2 (The fitness function is calculated using QDA)	5	98	100	95.83	96.30	100
	10	99.20	100	98.33	98.48	100
	15	97.25	97.60	96.88	97.13	97.38
	20	98	98.25	97.73	97.91	98.10
	25	98.43	98.63	98.21	98.36	98.51
Experiment-3 (The fitness function is calculated using KNN)	5	98.67	100	97.22	97.50	100
	10	100	100	100	100	100
	15	97.56	97.86	97.22	97.45	97.67
	20	98.17	98.40	97.92	98.08	98.26
	25	98.53	98.72	98.33	98.47	98.61

To better visualize the comparison of results, we have graphically represented the results of experiment 1, experiment 2 and experiment 3 in the figures 5.3, 5.4 and 5.5 respectively. In the first case of experiment 1, when the number of genes is chosen as 5, the value of performance measures are accuracy (98%), sensitivity (100%), specificity (95.83%), PPV (96.30%) and NPV (100%). In the second case, the number of genes chosen is 10; the experiment leaves us with values as accuracy (99%), sensitivity (100%), specificity (97.92%), PPV (98.11%) and NPV (100%). In case 3, after changing the number of genes to 15, the statistics found are accuracy (96.86%), sensitivity (97.25%), specificity (96.43%), PPV (96.72%) and NPV (97.01%). In the fourth case, when we change the number of genes to 20, the value of performance measures is accuracy (97.80%), sensitivity (98.08%), specificity (97.5%), PPV (97.70%) and NPV (97.91%). In last case when the number of genes is 25, the value of performance measures are accuracy (98.31%), sensitivity (98.52%), specificity (98.08%), PPV (98.23%) and NPV (98.39%). Among all the varying number of genes, the best performance is given by case 2 when the number of genes is taken to be 10. The results of experiment 1 are demonstrated in figure 5.3.

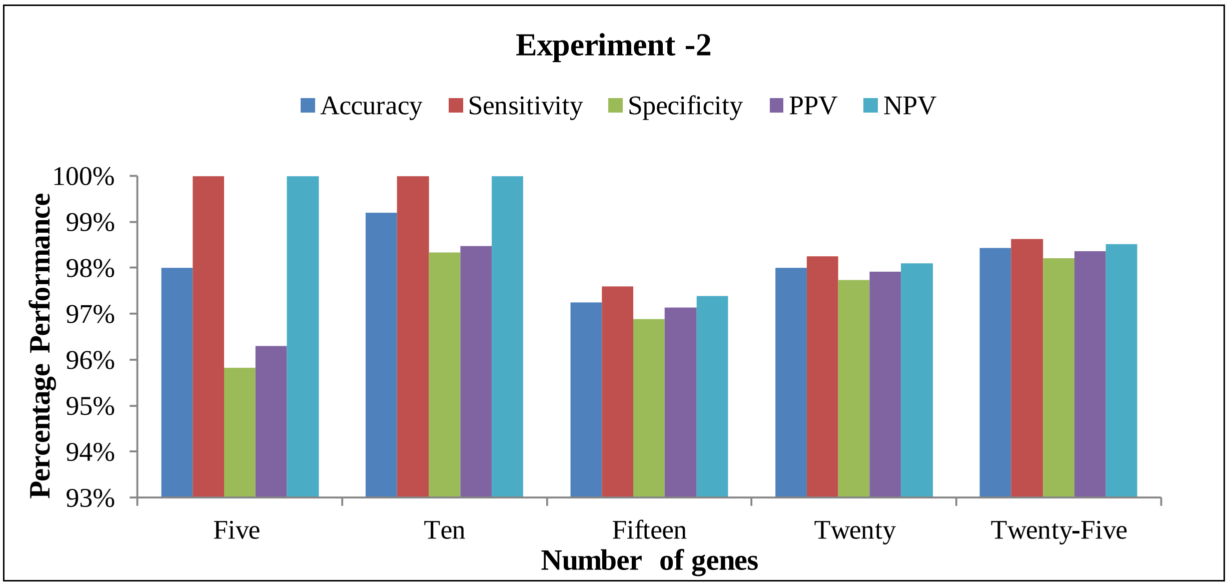
Figure 5.3: Fitness function is evaluated using LDA



In both experiments, namely experiment 1 and experiment 2, when the number of genes is chosen to be 5, the performance measures are exactly same. Here also, we have got the values as accuracy (98%), sensitivity (100%), specificity (95.83%), PPV (96.30%) and NPV (100%). In the second case, the number of genes is changed to 10; the results are accuracy (99.20%), sensitivity (100%), specificity (98.33%), PPV (98.48%) and NPV (100%). In the third case, when the number of genes is varied to 15; the outcome of this is accuracy (97.25%), sensitivity (97.60%), specificity (96.88%), PPV (97.13%) and NPV (97.38%). In the fourth case, where we have taken the number of genes to be 20, the performance measures are accuracy (98%), sensitivity (98.25%), specificity (97.73%), PPV (97.91%) and NPV (98.10%). In the last case, we change the number of genes to 25; the values are accuracy (98.43%), sensitivity (98.63%), specificity (98.21%), PPV (98.36%) and NPV (98.51%). Here also, among all the cases with varying number of genes, the value of performance measures are found to be best when the number of genes is taken 10. The results of experiment 2 are illustrated in figure 5.4.

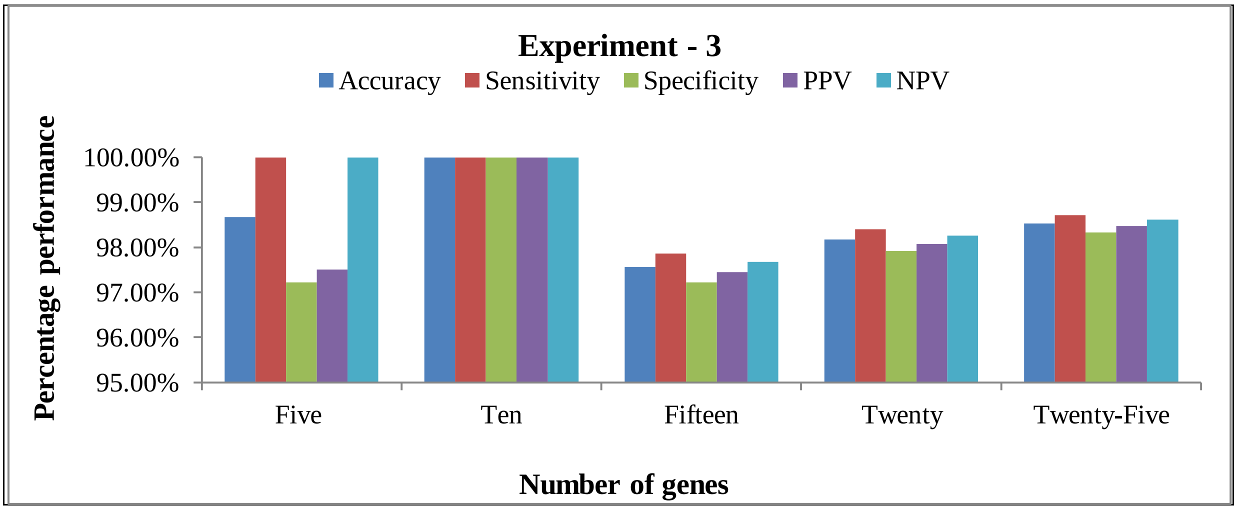
In experiment 3, for the first case when the number of genes is 5, the performance measures are accuracy (98.67%), sensitivity (100%), specificity (97.22%), PPV (97.50%) and NPV (100%). In the second case, the value of all performance measures is found to be 100% when we have taken the number of genes is 10. In the third case, the number of genes is changed to 15, the result of performance measures are accuracy (97.56%), sensitivity (97.86%), specificity (97.22%), PPV (97.45%) and NPV (97.67%). In the fourth case, the experiment leaves us with values of performance measures as accuracy (98.17%), sensitivity (98.40%), specificity (97.92%), PPV (98.08%) and NPV (98.26%). In the last case, the number of genes is changed to 25; the results are accuracy (98.53%), sensitivity (98.72%), specificity (98.33%), PPV (98.47%) and NPV (98.61%). The results of experiment 3 are shown in figure 5.5.

Figure 5.4: Fitness function is evaluated using QDA



In all three experiments, among all the varying number of genes, the best performance measures are found when the fitness function is calculated using KNN and the number of genes is taken to be 10. We have made an intelligent integrated method of gene selection for diagnosis of FSHD. The integrated technique selects a small subset of genes which will help in better diagnosis of FSHD and other NMDs also. Ultimately, the proper diagnosis will help in taking the accurate treatment option.

Figure 5.5: Fitness function is evaluated using KNN



5.4 Conclusions

The process of selecting the subset of discriminating genes for the genetic testing of NMDs is a crucial task. In this chapter, we have developed an integrated technique for gene selection and classification of FSHD. In the present work, a filter model, i.e., t-test is used to preselect the genes in the first stage and then in the second stage, GA as an embedded model with varying number of genes is used to select the most informative gene subset wherein the fitness function is calculated using LDA, QDA and KNN one-after-another. In the third stage, classification task is performed using the above mentioned classifiers. The comparison of results demonstrates that using KNN for evaluating the fitness function with 10 numbers of genes is found to be better as compared to the LDA and QDA with different number of genes. This method not only selects a small number of genes, but also improves the performance measures. We also intend to apply this integrated technique for gene selection and classification of other NMDs. Hence, the problem of gene selection for binary class classification is completely resolved as the proposed intelligent integrated approach has given the 100% accuracy with only a few numbers of genes. But the issue of gene selection for multi-class classification is still remains challenging.

The work presented in this chapter has been published in International Journal of Biomedical Engineering and Technology, Vol. 24, No. 3, pp. 285-296, 2017, Inderscience Publishers, United Kingdom, DOI: 10.1504/IJBET.2017.085144 mentioned under list of publications at the end of Chapter 8.

Chapter 6

A Novel Hybrid Feature Selection Model for Classification of Neuromuscular Dystrophies using Bhattacharyya Coefficient, Genetic Algorithm and Support Vector Machine

In this chapter, a multi-class classification of an NMD dataset is performed. A gene selection model is designed using the filter and embedded models of feature selection to overcome the issue of dimensionality reduction, and thus leading to the accurate classification of multi-class NMD datasets. From the filter models, bhattacharyya coefficient is chosen and GA as an embedded model is chosen and both of these are integrated. The results of integrated technique bhattacharyya-GA are compared with the individual techniques, i.e., bhattacharyya and GA. So, the chapter is structured as follows: Section 6.1 presents the introduction. Section 6.2 gives the methodology for gene selection and classification of NMDs. Section 6.3 presents the experimental results and their comparison. Section 6.4 concludes the chapter.

6.1 Introduction

The neuromuscular system in human body provides the vital forces to perform various actions [171]. The NMD occurs due to the mutation in the gene(s) that affects the motor unit. The symptoms of these diseases are progressive in nature. According to Muscular Dystrophy Foundation Australia, the genetic testing is used for diagnosis, which involves the direct examination of DNA associated with a particular kind of NMD. Usually, blood tests are used for genetic testing, which measures the level of certain enzymes in the blood. An accurate classification of NMDs is important in providing proper treatment facilities to the patients. These days, the microarray technology is used to analyze the whole genome simultaneously which monitor the level of activity or expression of a large number of genes [15] [120]. But the gene expression data derived from the microarray experiment usually involves a large number of genes, but only a very few number of samples [108]. Most of the genes in the samples do not contain useful information as they are redundant, not differentially expressed and are not specific to the disease. There is a need to reduce the dimension of gene expression data which intends to find a small set of discriminating genes that accurately classifies the samples of various kinds of diseases. Thus, reducing the dimension, i.e., number of genes (acting as features in machine learning) prior to the classification task will be helpful in accurately diagnosing a disease. It does not only increase the classification accuracy, but also decreases the computational burden [108]. So, our goal is to find a small subset of genes which ensures the accurate classification of NMDs.

The major aim of using feature selection procedure prior to diagnosis of any disease is to develop a diagnostic model based on the least possible number of genes. In the previous chapters, we have used feature selection methods for selection of genes for binary class classification datasets [166] [168] [175]. But, in this chapter, we propose a simple integrated model which first selects the features, and then classifies the samples of multi-class NMD datasets. Here, the process of feature selection is done in two phases by integrating Bhattacharyya coefficient and GA. In the first phase, Bhattacharyya coefficient forms a candidate feature subset which excludes the uninformative, redundant and noisy features. In the second phase, GA as an embedded model is deployed to find out the target feature subset which best discriminates the biological samples of different NMDs. The fitness function of the GA is calculated using the RBF SVM classifier. The results of the proposed integrated technique are compared with two individual techniques of feature selection, namely the Bhattacharyya coefficient and GA, and one integrated technique, i.e., Bhattacharyya-GA wherein the fitness function of GA is calculated using LDA, QDA, KNN and linear SVM. These individual and integrated techniques are applied on two different datasets of NMDs. The results show that the integrated technique Bhattacharyya-GA is found to be very effective for the classification of NMDs.

6.2 Methodology

In this section, we describe the methodology of the proposed integrated technique Bhattacharyya-GA for feature selection and classification of NMDs. The aim of this integrated technique is to select a small subset of informative features which best discriminates the biological samples of various diseases. Our proposed integrated method involves three steps. In step 1, the data is partitioned using the K-fold cross-validation technique into a training set and test set. The value of K is chosen to be 5. The detailed description of K-fold cross validation is given in section 2.7.

Step 2 consists of the two phases where we select the most discriminating features to make the target feature subset. In phase 1 of step 2, the Bhattacharyya coefficient is calculated from each feature and the top valued features forms the candidate gene subset. In phase 2 of step 2, the candidate feature subset is given as an input to GA to select the highly disease related features to form a target feature subset. In step 3, we input the selected target feature subset into our classification algorithms. In this methodology, different classification algorithms implemented are LDA, QDA, KNN, linear SVM and RBF SVM. The whole sequence of the overall process is shown in figure 6.1.

The process of feature selection selects a small subset of informative features which are highly related to the disease and are most important for their classification. Hence, the primary goal of feature selection is to identify important features responsible for classification. Bhattacharyya coefficient is used to determine the relative closeness of two samples. It is self-consistent, unbiased and applicable to any distribution [176]. It measures the amount of overlap between two samples and the seperability of classes in the classification [177]. Let us assume that C_m and D_m are number of members of samples C and D in the mth partition and N is the total number of partitions. The probability distribution is defined in equation 6.1.

$$\sum_{m=1}^{N} C_m = \sum_{m=1}^{N} D_m = 1 \tag{6.1}$$

The Bhattacharyya coefficient is calculated by equation 6.2

$$PQ(C,D) = \sum_{m=1}^{N} \sqrt{C_m D_m} \tag{6.2}$$

GA imitates the process of natural selection and belongs to the class of bio-inspired and evolutionary algorithms. The most important goal of GA is the continuous improvement in small steps for solving a problem by keeping the fittest amongst individuals over the repeated generations. The detailed description of GA is given in section 5.3. The GA parameters used in the present chapter are given in table 6.1.

Figure 6.1: Proposed Methodology

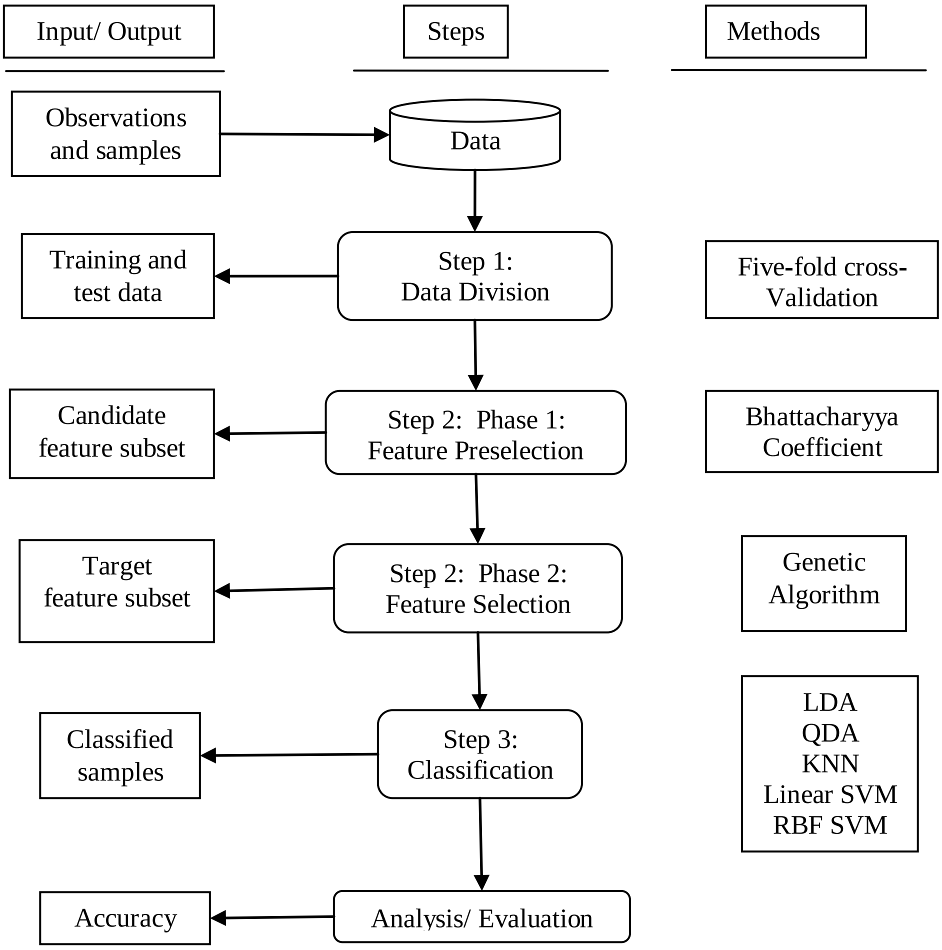


Table 6.1: GA Parameters

Parameter	**Value**
Chromosome	20 genes
Population Size	100 chromosomes
Selection Function	Stochastic Uniform Selection
Crossover Function	Scattered Crossover Function
Mutation Function	Gaussian Mutation Function
Termination Criteria	Value of fitness function is 100%

Now consider a NMD classification problem in which the gene expression value of z number of genes is given in the vector in equation 6.3 and the selector variables which label the class of NMD of tissue samples are given in equation 6.4.

$$a = (a_1, a_2, a_3 \cdots \cdots a_z) \in A \subseteq B^z \tag{6.3}$$

$$s \in S = \{1,2,3, \cdots \cdots B\} \tag{6.4}$$

For the classification of samples, five classification algorithms namely LDA, QDA, KNN, linear SVM and RBF SVM are used given in subsections 2.5.1-2.5.4. SVM supports only binary class classification problem. In case of multiclass classification problem, i.e., more than two classes in the dataset, binary SVM are not sufficient for classification of samples of whole dataset. So, we have to reduce the single multi-class (K classes) problem into K binary classification problems. In the present chapter, the OVA approach with linear SVM and RBF SVM is used for classification of NMDs. In this case, if there are K numbers of classes in the whole dataset, then K binary classifiers are built where each binary classifier picks out one class from all the other classes.

6.3 Results and Discussion

We evaluated the performance of our proposed hybrid model on two publicly available datasets of NMDs. The datasets are taken from experiment E-GEOD-3307 and we named these datasets as DST-4 and DST-5 and are detailed in subsections 2.3.4 and 2.3.5 respectively [153]. Both of the datasets contain a small number of samples and a large number of features due to which it is very difficult to accurately classify the samples of these datasets. The accuracies of all the individual and integrated techniques on two microarray datasets of NMDs DST-4 and DST-5 are tested and compared. The result of the experiment 1 is shown in table 6.2 which gives the classification accuracies of both the data sets using LDA, QDA, KNN, Linear SVM and RBF SVM classifiers calculated when individual technique Bhattacharyya coefficient is employed for feature selection. From this, it is observed that the classification accuracies are not so good, especially from LDA, QDA and KNN classifiers. The linear and RBF kernels of SVM have given slightly better classification accuracies as compared to other classifiers.

Table 6.2**:** Percentage classification accuracy with first 100 selected genes using Bhattacharyya

Data	**LDA**	**QDA**	**KNN**	**Linear SVM**	**RBF SVM**
DST-4	36.11	50	61.11	73.61	74.88
DST-5	41.82	56.97	65.45	77.54	82.54

Experiment 2 is conducted in which the first 100 genes are selected using GA and the classification accuracies are calculated using LDA, QDA, KNN, linear SVM and RBF SVM. The results of this experiment are shown in table 6.3. In the first phase of experiment 3, Bhattacharyya coefficient is used to form the candidate gene subset. In the second phase, GA is employed to find the target gene subset and the classification is done using LDA, QDA, KNN, linear SVM and RBF SVM. Table 6.4 demonstrates the results of experiment 3. To summarize, the proposed integrated technique Bhattacharyya-GA, wherein the fitness function of the GA is calculated using RBF SVM exhibits better performance on both of the data sets.

Table 6.3 Percentage classification accuracy with first 100 selected genes using GA

Data	**LDA**	**QDA**	**KNN**	**Linear SVM**	**RBF SVM**
DST-4	59.72	60.69	71.48	88.61	90.92
DST-5	68.78	69.39	78.38	89.45	89.84

Table 6.4 Percentage classification accuracy with first 100 selected genes using Bhattacharyya-GA

Data	**LDA**	**QDA**	**KNN**	**Linear SVM**	**RBF SVM**
DST-4	82.11	86.69	93.498	97.988	**98.464**
DST-5	81.13	86.21	93.47	98.1	**98.48**

Due to the curse of a large number of genes in microarrays as compared to a small number of samples, the classification algorithms endure the high dimensional input space problem. This high dimensionality of microarray data degrades the classification accuracy and increases the computational complexity. Thus, an optimal subset of genes, i.e., 100 genes, is required for the classification of these diseases while maintaining the good classification accuracy. Besides of choosing this less number of genes for higher accuracy, it is also possible to interpret their gene expression profile for further drug discovery. The proposed integrated technique not only maximizes the classification performance, but also maintains the computational complexity. It selects the less number of genes required for the accurate classification which leads to less computational time for processing and less computational cost.

6.4 Conclusions

In the past decade, microarray technology has led a huge impact on the cancer research and has given great results in prediction of cancer classes. So, in the present chapter, we use this technology for feature selection and classification of NMD datasets. We made an intelligent integrated technique Bhattacharyya-GA for feature selection. Bhattacharyya coefficient is calculated to make a candidate gene subset by excluding redundant and uninformative genes in the first phase. In the second phase, GA is applied to find out the most informative gene subset to form target gene subsets. The fitness function of the GA is computed using different classification algorithms, namely LDA, QDA, KNN, linear SVM and RBF SVM. The novel hybrid feature selection model Bhattacharyya-GA is applied on two huge microarray datasets and the performance is compared with Bhattacharyya and GA alone. The experimental results show that the proposed technique Bhattacharyya-GA, when the fitness function is calculated using RBF SVM has outperformed on both data sets.

We have shown that the proposed integrated technique gives highly encouraging classification accuracies and hence it can be applied for classification of other NMDs also. Even if the proposed integrated technique has given the high value of performance measures, still we can try some other methods which not only increase the classification accuracy but also decreases the number of genes required to achieve that classification accuracy.

The work presented in this chapter has been published in Interdisciplinary Sciences: Computational life Sciences, Vol. 10, No. 2, 244-250, Springer Berlin Heidelberg, Germany, DOI: 10.1007/s12539-016-0183-6.

Chapter 7

A Novel Approach for Dissimilar Gene Selection and Multi-Class Classification of Neuromuscular Disorders

In this chapter, we try to select very less number of dissimilar genes for every class which helps in getting the high classification accuracy. The genes are selected using the median matrix formed by the processing of gene expression matrix. The details are given in this chapter. So, the chapter is structured as follows: Section 7.1 presents the introduction. Section 7.2 gives the methodology for selection of compact subsets of genes and multi-class classification of NMDs. Section 7.3 presents the experimental results and their comparison. Section 7.4 concludes the chapter.

7.1 Introduction

The neuromuscular system consisting of the nervous system and muscular system provides the vital forces to perform various actions. The mutation in one or more genes damages the DNA, which changes the mechanism of cell replication and causes the NMD. These disorders affect the peripheral nervous system and muscles which leads to different levels of severity varying from minor loss of strength or numbness to muscle death or paralysis [171]. A person suffering from NMD is not able to perform the voluntary movements. So, an accurate prediction of the kind of NMDs is fundamental for choosing the optimal treatment for patients. The monitoring of gene expression data through microarrays leads to the proper classification [179].

Thus, for the accurate diagnosis of these disorders, the microarray technology came into the picture. The microarray technology focuses on finding the gene expression level of a large number of genes simultaneously in an experiment. The design of microarray chip contains a microscopic ordered array which holds the genotype of all the genes. This technology is powerful for providing the useful diagnostic information by investigating and comparing the differences between gene expression profiles of healthy and diseased samples under a particular condition. It is used to identify the subsets of differentially expressed biomarker genes responsible for some kind of disorder development. It gives a more reliable and accurate way of diagnosing the disorders.

Unfortunately, the microarray data of NMDs are cursed from high dimensionality as it contains the expression profiles of tens of thousands of genes. From all these examined genes, only a few genes are significantly appropriate under a particular condition which can be considered as a subset of biomarker genes. Rest all the genes are inappropriate, irrelevant, redundant and noisy, which deteriorates the classification performance and increases the experimental cost. Thus, there is a need to select a few biomarker genes to accurately classify these diseases. The objectives of gene selection are fourfold: 1) enhancing classification performance by removing redundant genes, 2) reducing the computational burden of the classifier by excluding noisy genes, 3) cutting down the cost of genetic testing of NMDs by selecting only informative genes and 4) providing a path to further investigate the relationships between selected genes and treatment of these diseases. Then the patient will be in a better position to prevent or cure these diseases. Here we intend to select the most compact subsets of dissimilar and discriminating genes from thousands of genes that can successfully classify the various kinds of NMDs.

But it is extremely difficult to select these genes from thousands of genes due to the presence of only few samples of each type of NMDs. In order to confirm the validity of these selected biomarker genes, biologically associated with a particular kind is to check the classification accuracy of the sample classifier built using these selected genes. Thus, we propose a new integrated model for gene selection and multi-class classification of NMDs. The gene expression matrix is processed to create a median matrix for the selection of compact and different subsets of genes for every class. The classification algorithms use the combination of these selected genes for prediction of the kind of NMD samples. The various classification algorithms employed are LDA, QDA, KNN, linear SVM and RBF SVM. Our technique uses the OVA approach to decompose the multi-class classification problem into binary class classification problem. The accuracy and effectiveness of the proposed model are exhibited through analysis of publicly available microarray dataset of 13 NMDs. It selects only a few biomarker genes for each class of NMD. It selects a minimum of 4 genes in one class and a maximum of 19 genes in another class. The integration of the proposed method of gene selection with RBF SVM classification algorithm has outperformed in most of the cases. The results confirm the ability of the proposed gene selection and classification model for identifying the subsets of most discriminating and non-redundant genes which helps the classifier to give a high classification performance.

From the literature, it was observed that a huge amount of work has been done on gene selection and classification of cancer and tumor data sets. But almost negligible amount of work has been found in the gene selection in NMDs data sets. So, the selection of discriminating genes for accurate classification of NMDs into different kinds is highly needed for choosing the optimal treatment option for patients. The challenge here is that almost all the gene selection and classification methods work on the data sets which contains only two classes. But the task of selecting biomarker genes for classification of multi-class datasets is still restricted. Thus, in the present chapter, we propose an integrated model for gene selection and multi-class classification of NMDs which selects only a few genes for each class to give higher classification accuracies. Here, the genes are selected using a median matrix which is created using gene expression matrix. Further, the combination of selected genes is used to construct classifiers. The data set consists of 13 classes of NMDs is taken from The European Molecular Biology Laboratory - European Bioinformatics Institute (EMBL-EBI). The genes selected using the proposed model has significantly increased the classification accuracies. Thus, this leads to the accurate classification by considering the gene expression levels of only few selected genes. In the previous chapter, the classification of multi-class NMD data sets is done by selecting 100 genes [180].

7.2 Methodology

We sought to develop an integrated method for the classification of NMDs based on those selected genes whose expression is particular to each kind of NMD. For the evaluation, the data set DST-6 was taken from EMBL-EBI and the detailed description is given in section 2.3. The overall structure of the proposed methodology is given in figure 7.1. We have a multi-class NMDs microarray dataset which contains *A* classes and *B* samples of *C* genes. The gene expression matrix signifies the expression levels of genes in samples where each row represents a gene and each column represents a sample. The value d_{ef} represents the expression value of gene *e* in sample *f*. The task here is to categorize the samples into their respective classes based upon the gene expression values in the samples. The overall procedure encompasses of four steps which are discussed below.

1. Data preprocessing 2. Creation of a median matrix 3. Gene ranking and selection 4. Building the classifier.

Data preprocessing

The data set contains 13 classes which need to be preprocessed before presenting to the algorithm. Here, each instance must be represented in the form of a real number. So the categorical values of the classes should be converted into a real number. Hence, class AQM is assigned a value 1, ALS is assigned a value 2 and so on. The classes are represented by number 1 to 13.

Creation of a median matrix

After preprocessing the data, the median matrix from the gene expression matrix is created [181]. The samples of each class are separated from each other. For all eth genes and gth classes, the median of expression levels is calculated and represented as M_{eg}, for e = 1, 2, 3 …. 22645 and g = 1, 2, 3…13. For the eth gene, the differences of medians of expression levels of each class from every other class are calculated. It is given in equation 7.1

$M_e = M_{e1}$-M_{e2}, M_{e1}-M_{e3}, M_{e1}-M_{e4}………M_{e1}-M_{e13}, M_{e2}-M_{e3}, M_{e2}-M_{e4}, M_{e2}-M_{e5}............M_{e2}-M_{e13}, M_{e3}-M_{e4}, M_{e3}-M_{e5}, M_{e3}-M_{e6}………M_{e3}-M_{e13}, M_{e4}-M_{e5}, M_{e4}-M_{e6}, M_{e4}-M_{e7}………M_{e4}-M_{e13}, M_{e5}-M_{e6}, M_{e5}-M_{e7}, M_{e5}-M_{e8}………M_{e5}-M_{e13}, M_{e6}-M_{e7}, M_{e6}-M_{e8}, M_{e6}-M_{e9}………M_{e6}-M_{e13}, M_{e7}-M_{e8}, M_{e7}-M_{e9}, M_{e7}-M_{e10}………M_{e7}-M_{e13}, M_{e8}-M_{e9}, M_{e8}-M_{e10}, M_{e8}-M_{e11}………M_{e9}-M_{e10}, M_{e9}-M_{e11}, M_{e9}-M_{e12}, M_{e9}-M_{e13}, M_{e10}-M_{e11}, M_{e10}-M_{e12}, M_{e10}-M_{e13}, M_{e11}-M_{e12}, M_{e12}-M_{e13}. (7.1)

We have taken the absolute value of all the differences of medians. The total number of columns is equal to the g (g-1)/2 for *g* classes. Next is to place all the 0s at the end of the matrix to make it equal to the size of gene expression matrix. The size of median matrix and the gene expression matrix must be equal in order to classify the samples to their classes.

Gene ranking and selection

We assign a rank to each gene in the training data set which describes its capability of classifying a class from the rest of the class. We run a ranking method for all the genes of a class which ranks and then sorts the genes in descending order of their ranks. The gene with the highest rank is assigned as Gene 1; the gene with the second highest rank is assigned as Gene 2 and so on. In the present chapter, the ranking algorithm employed is the t-test explained in section 4.2. The genes are selected from the training data set with the following procedure: While moving through the ordered genes in first run, we selected only one gene from the list with the highest score labeled as Gene 1 in table 1(a), 1(b) and put that to biomarker gene subset. We use only this gene, i.e., Gene 1 to classify a single class from all other classes in every case using various classification algorithms. During the first run, we added only one selected gene with the highest rank (labeled as Gene 1 in Table 1(a), 1(b)) to the biomarker gene subset and then used this only gene for classification. The PMs are calculated using all classification algorithms under the LOOCV scheme. If we did not find the satisfactory classification accuracy, then the algorithm moves to the second run, where it adds the next gene with the second highest score (labeled as Gene 2 in Table 1(a), 1(b)) to the biomarker gene subset for classification. Now we have two genes in our biomarker gene subset and we keep on adding one gene to the subset in every run till we got the excellent classification performance. Here, we have chosen different biomarker genes for every class.

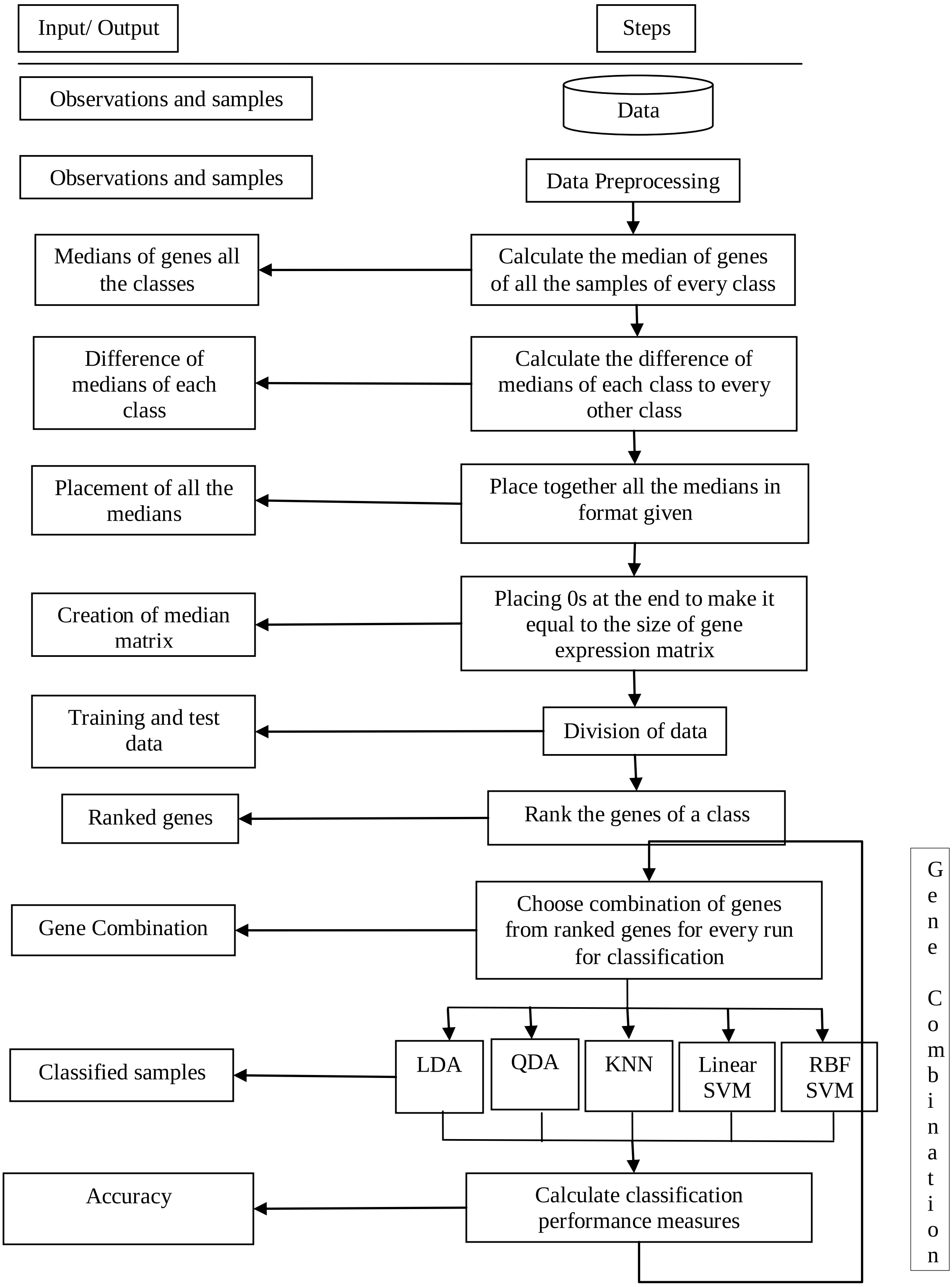
Building the classifier

The quality of selected genes is evaluated by using them in building the classifiers. The selected genes of every class are fed into classification algorithms one-by-one. For the classification of multi-class datasets, the binary classifiers are not sufficient. So, the idea of binary class classification is extended to multi-class classification using OVA approach. The whole problem is divided into n binary classification problems [182], where each binary classifier picks out one particular class from rest of the classes. It considers the samples of that class as positive and the samples of rest of the classes as negative. To categorize a sample to its respective class, all the binary classifiers are used. The one who receives the strongest prediction or maximizes the decision function, the sample will be assigned to that class. The decision function used is given in equation 7.2

$$x_{\theta}{}^{(n)}{}_{(r)} = p(y = n|r;\theta) \tag{7.2}$$

We have employed five classification algorithms, namely LDA, QDA, KNN, linear SVM and RBF SVM one-by-one to classify the samples into their respective classes explained in section 2.5.

Figure 7.1: Overall Structure


Input/ Output
Steps
Observations and samples
Data
Observations and samples
Data Preprocessing
Medians of genes all the classes
Calculate the median of genes of all the samples of every class
Difference of medians of each class
Calculate the difference of medians of each class to every other class
Placement of all the medians
Place together all the medians in format given
Creation of median matrix
Placing 0s at the end to make it equal to the size of gene expression matrix
Training and test data
Division of data
Ranked genes
Rank the genes of a class
Gene Combination
Choose combination of genes from ranked genes for every run for classification
Gene Combination
Classified samples
LDA
QDA
KNN
Linear SVM
RBF SVM
Accuracy
Calculate classification performance measures

7.3 Experimental Results and Discussion

In this section, we present the evaluation method used to evaluate the performance of the proposed integrated technique. Here, LOOCV technique is used to estimate the generalization performance of the model. LOOCV technique's complete description is given in section 2.7. The quantitative performance measure (PM) calculated is ACC of the classifiers according to the chosen number of genes as mentioned in section 2.6.

The process of selecting biomarker genes from a large data set specific to the kind of NMD reflects a high level of difficulty. The gene selection procedure employed in the present chapter gives the subsets of important genes for every class which guarantees the accurate classification of various kinds of NMDs. The biomarker subsets are empty at the initial stage. The final subsets of biomarker genes of every class contain only those informative genes which benefit the classifiers in accurately classifying the dataset. Under the LOOCV scheme, the minimum number of genes chosen was 4 in one class and maximum number of genes chosen was 19 in another class, from 22,645 genes. The IDs of important genes of every class are given in tables 7.1 (a) and 7.1 (b).

Table 7.1 (a)**:** Biomarker genes

Genes	**Class**						
	AQM	**ALS**	**BMD**	**FSHD**	**JD**	**DMD**	**SPG4**
Gene 1	20648	3379	9346	10490	17991	4557	13394
Gene 2	4399	9675	4847	10752	10055	16958	11153
Gene 3	1037	19368	9319	21244	11195	15206	9053
Gene 4	18808	12414	12964	13514	5885	6431	17140
Gene 5	NA	10188	8730	10980	2324	20050	2693
Gene 6	NA	1549	12666	13675	9222	219	195
Gene 7	NA	16539	2048	12314	21448	16141	10313
Gene 8	NA	21618	671	8195	4334	17041	18504
Gene 9	NA	2705	16513	7400	6757	13466	21278
Gene10	NA	17364	4284	11147	6656	8628	21602
Gene11	NA	NA	1549	17648	8773	2529	9072
Gene12	NA	NA	5637	15419	19060	4124	11973
Gene13	NA	NA	4649	17237	10580	NA	6639

Gene14	NA	NA	16504	8512	3184	NA	5897
Gene15	NA	NA	7784	NA	3304	NA	NA

Table 7.1 (b)**:** Biomarker genes

Genes	**Class**					
	AD-EDMD	**X-Linked – EDMD**	**Caplain-3**	**Dysferlin**	**FKRP**	**NHSM**
Gene 1	13124	13624	1839	9754	9754	9754
Gene 2	19000	20457	2673	12915	12915	12915
Gene 3	21695	11702	4293	12780	12780	12780
Gene 4	20818	6115	12240	18869	18869	18869
Gene 5	4662	13855	NA	15000	15000	15000
Gene 6	322	20733	NA	22561	22561	22561
Gene 7	5786	11420	NA	14207	14207	14207
Gene 8	14286	1578	NA	14090	14090	14090
Gene 9	13819	12196	NA	NA	2319	2319
Gene10	NA	14285	NA	NA	16849	16849
Gene11	NA	20295	NA	NA	7122	7122
Gene12	NA	10347	NA	NA	2196	2196
Gene13	NA	5408	NA	NA	17597	17597
Gene14	NA	13152	NA	NA	9497	9497
Gene15	NA	12217	NA	NA	14738	14738
Gene16	NA	4834	NA	NA	4684	4648
Gene17	NA	13447	NA	NA	NA	2951
Gene18	NA	NA	NA	NA	NA	18671
Gene19	NA	NA	NA	NA	NA	6983

The best LOOCV classification accuracies using only those genes in both training and test data sets are also calculated. Tables 7.2-7.14 show the performance measure of all the experiments of every class. As shown in table 7.2, which illustrates the experiment with the AQM dataset, LDA achieves the best classification accuracy in the training data set (100%) and in the test data set (98.35%) using only 4 genes. On the other hand, other classifiers have also performed well. The ALS dataset has 10 selected genes. Here also, LDA achieved the highest classification accuracies of the training data set (97.22%) and of the test data set

(95.87%) which is illustrated in table 7.3. In case of BMD data set, RBF SVM achieved the highest classification accuracies in both training data set (90.54%) and test data set (90.58%) with only 15 genes which is demonstrated in table 7.4. While LDA and QDA achieved very less accuracy in this data set. The FSHD data set has 14 genes selected. As seen in table 7.5, here also RBF SVM achieved highest classification accuracies in both training data set (84.80%) and test data set (84.96%). However, linear SVM has also performed well as compared to LDA, QDA and KNN with the same number of genes.

Table 7.2: LOOCV Performance measures for AQM data set during different runs

Run	**Classification algorithms**									
	LDA		**QDA**		**KNN**		**Linear SVM**		**RBF SVM**	
	TR	**TS**	**TR**	**TS**	**TR**	**TS**	**TR**	**TS**	**TR**	**TS**
1	93.52	91.74	92.14	91.32	94	93.66	94.27	94.21	94.59	94.55
2	98.15	96.69	94.76	94.21	96	95.59	96.18	96.07	96.45	96.36
3	98.15	97.52	96.51	95.87	97.14	96.97	97.45	97.31	97.8	97.69
4	100	98.35	97.38	96.69	97.71	97.52	98.08	97.73	98.14	98.02

Table 7.3: LOOCV Performance measures for ALS data set during different runs

Run	**Classification algorithms**									
	LDA		**QDA**		**KNN**		**Linear SVM**		**RBF SVM**	
	TR	**TS**	**TR**	**TS**	**TR**	**TS**	**TR**	**TS**	**TR**	**TS**
1	87.96	88.43	89.08	89.26	92.57	92.56	92.36	92.36	92.91	92.89
2	87.96	88.43	86.03	85.95	90.29	90.63	90.87	90.91	91.89	91.9
3	87.96	88.43	86.03	85.54	90	90.36	90.87	90.91	91.89	91.9
4	86.11	86.78	85.59	85.12	89.71	90.08	90.66	90.7	91.72	91.74
5	86.11	85.95	85.59	85.12	89.71	90.08	90.87	90.91	91.89	91.9
6	92.59	91.74	88.65	84.62	92	92.01	92.99	92.77	93.41	93.22
7	92.59	91.74	90.39	90.08	93.14	92.84	93.42	93.18	94.26	94.21
8	92.59	91.74	90.83	90.5	93.43	93.39	94.06	93.8	94.76	94.71
9	96.3	94.21	92.14	91.74	94.29	94.21	94.9	94.63	95.61	95.54
10	97.22	95.87	93.01	92.98	95.14	95.04	95.54	95.45	95.78	95.7

Table 7.4: LOOCV Performance measures for BMD data set during different runs

Run	Classification algorithms									
	LDA		QDA		KNN		Linear SVM		RBF SVM	
	TR	TS	TR	TS	TR	TS	TR	TS	TR	TS
1	38.89	38.84	48.03	47.93	64	65.01	72.19	72.73	77.03	77.36
2	43.52	42.98	55.9	56.2	69.71	70.25	76.22	76.65	80.24	80.5
3	51.85	49.59	61.57	61.98	73.71	74.1	79.19	79.55	82.6	82.81
4	53.7	52.07	59.39	59.5	72	72.45	77.92	78.31	81.59	81.82
5	57.41	55.37	62.01	61.98	73.71	74.38	79.41	79.75	82.77	82.98
6	74.07	73.55	71.62	71.49	80.29	80.44	84.08	84.3	86.49	86.61
7	76.85	76.03	71.18	71.49	80.29	80.44	84.08	84.3	86.49	86.61
8	75	73.55	69.87	70.25	79.43	79.61	83.44	83.68	85.98	86.12
9	75	76.03	70.74	71.49	80.29	80.44	84.08	84.3	86.49	86.61
10	75.93	76.86	71.62	72.31	80.86	81.27	84.71	84.92	86.99	87.11
11	76.85	77.69	72.49	73.14	81.43	81.82	85.14	85.33	87.33	87.44
12	76.85	77.69	72.49	73.14	81.43	81.82	85.14	85.33	87.33	87.44
13	81.48	79.34	77.73	78.1	84.86	85.12	87.69	87.81	89.7	89.75
14	75	75.21	76.42	76.86	84	84.3	87.26	87.19	89.19	89.26
15	83.33	80.99	79.48	79.75	86	86.23	88.96	88.84	90.54	90.58

Table 7.5: LOOCV Performance measures for FSHD data set during different runs

Run	Classification algorithms									
	LDA		QDA		KNN		Linear SVM		RBF SVM	
	TR	TS	TR	TS	TR	TS	TR	TS	TR	TS
1	59.26	57.85	52.84	52.48	67.14	67.77	72.4	72.93	75.68	76.03
2	56.48	57.02	55.9	54.96	68.86	69.7	73.89	74.38	76.86	77.19
3	59.26	58.68	48.47	46.69	63.14	63.91	69.43	70.04	73.31	73.72
4	56.48	56.2	51.53	50.41	65.71	66.39	71.34	71.9	74.83	75.21
5	65.74	65.29	57.21	55.79	69.43	69.97	74.1	74.59	77.03	77.36
6	62.96	62.81	56.33	54.55	68.57	69.15	73.46	73.97	77.03	77.36
7	65.74	65.29	58.95	57.44	70.57	71.35	75.16	75.62	79.05	79.34
8	68.52	66.94	63.76	61.98	73.71	74.38	77.49	77.89	80.91	81.16
9	65.74	65.29	62.88	61.57	73.43	74.1	77.28	77.69	81.08	81.32
10	71.3	68.6	65.5	64.05	75.14	75.48	78.34	78.72	82.43	82.64

11	74.07	71.07	67.69	66.12	76.57	76.86	79.41	79.75	83.45	83.64
12	79.63	77.69	71.62	69.83	79.14	79.34	81.32	81.4	84.8	84.96
13	78.7	76.86	71.62	70.25	79.43	79.61	81.53	81.61	84.97	85.12
14	78.7	76.86	70.74	69.42	78.86	79.34	81.32	81.4	84.8	84.96

Table 7.6 shows the results of JD dataset. The best classification accuracy is achieved in the training data set (97.13%) and test dataset (97.02%) with RBF SVM using only 15 genes. Here, QDA, KNN and linear SVM have performed equally well. But LDA has performed worse in case of this dataset. The DMD data set has 12 genes selected; here again RBF SVM has achieved the best classification accuracy in the training data set (97.13%) as well as in the test dataset (97.02%) as seen in table 7.7. Table 7.8 demonstrates the results of SPG4 dataset achieved using 14 genes; the maximum classification accuracy is accomplished using RBF SVM in the training data set (97.13%) and test dataset (97.02%). For AD-EDMD dataset, the results are shown in table 7.9. The best classification accuracy in the training data set (94.26%) and test dataset (94.21%) is achieved using 9 genes by employing RBF SVM. Here LDA has performed slightly worse as compared to QDA, KNN and linear SVM classification algorithms.

Table 7.6: LOOCV Performance measures for JD data set during different runs

Run	**Classification algorithms**									
	LDA		**QDA**		**KNN**		**Linear SVM**		**RBF SVM**	
	TR	**TS**	**TR**	**TS**	**TR**	**TS**	**TR**	**TS**	**TR**	**TS**
1	77.78	77.69	76.42	76.45	83.71	84.02	83.86	84.09	83.78	83.8
2	78.7	79.34	79.48	79.75	86	85.95	85.77	85.74	85.98	86.12
3	79.63	80.17	80.35	80.58	86.57	86.5	86.41	86.36	86.82	86.94
4	82.41	82.64	82.1	82.23	87.71	87.6	87.69	87.6	88.18	88.1
5	83.33	83.47	83.84	83.88	88.86	88.98	88.96	88.84	89.7	89.75
6	88.89	89.26	89.96	90.08	93.14	93.11	92.36	92.56	93.58	93.55
7	87.04	87.6	88.21	88.43	92	92.01	92.14	92.36	93.58	93.55
8	84.26	85.12	88.21	88.43	92	92.01	91.72	91.94	93.41	93.39
9	84.26	85.12	88.65	88.84	92.29	92.01	91.72	91.94	93.41	93.39
10	84.26	85.12	89.96	90.8	93.14	93.11	92.78	92.98	94.26	94.17
11	87.96	88.43	91.7	91.74	94.29	94.21	93.63	93.8	94.93	94.88

12	88.89	89.26	92.58	92.56	94.86	94.77	94.48	94.63	95.61	95.54
13	90.74	90.91	94.32	94.21	96	95.59	95.12	95.25	96.11	96.03
14	91.67	92.56	95.63	95.45	96.86	96.42	95.97	95.87	96.62	96.53
15	91.67	92.56	96.07	95.87	97.14	96.97	96.6	96.49	97.13	97.02

Table 7.7: LOOCV Performance measures for DMD data set during different runs

Run	**Classification algorithms**									
	LDA		**QDA**		**KNN**		**Linear SVM**		**RBF SVM**	
	TR	**TS**	**TR**	**TS**	**TR**	**TS**	**TR**	**TS**	**TR**	**TS**
1	83.33	83.47	83.41	83.47	88.57	88.71	89.17	89.26	89.7	89.75
2	84.26	85.12	84.28	84.3	89.14	89.26	90.02	90.08	90.88	90.91
3	81.48	82.64	84.72	85.12	89.71	89.81	90.87	90.91	91.89	91.9
4	91.67	91.74	92.58	92.98	95.14	94.77	94.48	94.42	94.93	94.88
5	89.81	90.08	93.01	92.98	95.14	95.04	94.69	94.63	95.27	95.21
6	91.67	91.74	94.32	94.21	96	95.87	95.75	95.87	96.45	96.36
7	91.67	91.74	94.32	94.21	96	95.87	95.97	96.07	96.79	96.69
8	91.67	91.74	95.2	94.63	96.29	96.14	96.18	96.28	96.96	96.86
9	90.74	90.91	93.01	93.39	95.43	95.32	95.75	95.87	96.62	96.53
10	89.81	90.08	91.27	91.47	94.29	94.21	95.12	95.25	96.11	96.03
11	91.67	91.74	92.58	92.98	95.14	95.04	95.97	96.07	96.79	96.69
12	93.52	93.39	93.45	93.8	95.71	95.59	96.39	96.49	97.13	97.02

Table 7.8: LOOCV Performance measures for SPG4 data set during different runs

Run	**Classification algorithms**									
	LDA		**QDA**		**KNN**		**Linear SVM**		**RBF SVM**	
	TR	**TS**	**TR**	**TS**	**TR**	**TS**	**TR**	**TS**	**TR**	**TS**
1	41.67	42.98	35.81	35.54	55.43	56.75	66.03	66.74	72.3	72.73
2	47.22	48.76	49.78	50.83	66	66.94	73.89	74.38	78.55	78.84
3	53.7	55.37	58.95	59.92	72.29	73	78.56	78.93	82.26	82.48
4	89.81	88.43	78.17	78.1	84.86	84.85	87.69	87.81	89.53	89.59
5	88.89	87.6	76.86	76.86	84	84.02	87.05	87.19	89.02	89.09
6	88.89	87.6	85.59	85.95	90.29	90.08	91.72	91.74	92.74	92.73
7	82.41	80.99	85.59	85.95	90.29	90.36	91.93	91.94	92.91	92.89
8	84.26	84.3	87.34	87.6	91.43	91.46	92.78	92.77	93.58	93.55

9	83.33	83.47	86.9	87.19	91.14	91.18	92.57	92.56	93.58	93.55
10	89.81	87.6	90.39	90.5	93.43	93.39	94.48	94.42	95.27	95.21
11	90.74	88.43	91.27	91.32	94	93.94	95.12	95.04	95.78	95.7
12	94.44	91.74	93.89	93.8	95.71	95.59	96.18	96.07	96.79	96.69
13	94.44	91.74	93.89	93.8	95.71	95.59	96.39	96.28	96.96	96.86
14	95.37	92.56	94.3	94.21	96	95.59	96.6	96.49	97.13	97.02

Table 7.9: LOOCV Performance measures for AD-EDMD data set during different runs

Run	**Classification algorithms**									
	LDA		**QDA**		**KNN**		**Linear SVM**		**RBF SVM**	
	TR	**TS**	**TR**	**TS**	**TR**	**TS**	**TR**	**TS**	**TR**	**TS**
1	80.56	80.17	82.53	82.23	87.71	87.88	89.81	89.88	91.05	91.07
2	77.78	76.86	79.91	79.75	86	85.95	88.32	88.43	89.86	89.92
3	78.7	77.69	82.1	81.82	87.43	87.6	89.6	89.67	90.88	90.91
4	78.7	77.69	80.79	80.58	86.57	86.78	88.96	89.05	90.37	90.41
5	82.41	81.82	83.84	83.47	88.57	88.15	90.02	90.08	91.22	91.24
6	81.48	80.17	86.03	85.54	90	90.08	91.51	91.53	92.91	92.89
7	86.11	85.95	89.08	88.43	92	92.01	92.99	92.98	93.92	93.88
8	85.19	82.64	86.9	86.36	90.57	90.63	91.93	91.94	93.07	93.06
9	86.11	85.12	89.52	89.26	92.57	92.56	93.21	92.98	94.26	94.21

Table 7.10: LOOCV Performance measures for X-Linked-EDMD data set during different runs

Run	**Classification algorithms**									
	LDA		**QDA**		**KNN**		**Linear SVM**		**RBF SVM**	
	TR	**TS**	**TR**	**TS**	**TR**	**TS**	**TR**	**TS**	**TR**	**TS**
1	54.63	53.37	55.46	55.79	69.43	70.25	76.65	77.07	80.91	81.16
2	53.7	54.55	57.64	57.85	70.86	71.63	77.71	78.1	81.76	81.98
3	56.48	57.85	61.14	61.57	73.43	74.1	79.62	79.96	83.28	83.47
4	55.56	57.02	62.45	63.64	74.86	75.48	80.68	80.99	84.12	84.3
5	56.48	57.85	69.87	70.25	79.43	79.89	84.08	84.3	86.82	86.94
6	74.07	75.21	79.91	79.75	86	86.23	88.96	89.05	90.71	90.74
7	82.41	82.64	84.28	83.88	88.86	88.98	91.08	91.12	92.4	92.4
8	77.78	77.69	85.15	85.54	90	90.08	91.93	91.94	93.07	93.06

9	80.56	80.99	87.34	87.19	91.14	91.18	92.78	92.77	93.75	93.72
10	92.59	91.74	94.32	94.21	96	95.87	96.6	96.28	96.79	96.69
11	91.67	90.91	93.89	93.8	95.71	95.32	96.18	96.07	96.62	96.53
12	89.81	89.26	93.45	93.39	95.43	95.04	95.97	95.87	96.62	96.53
13	91.67	90.91	94.32	94.21	96	95.87	96.6	96.49	97.13	97.02
14	93.52	91.74	94.76	94.63	96.29	96.14	96.82	96.69	97.3	97.19
15	93.52	92.56	95.2	95.04	96.57	96.42	97.03	96.9	97.47	97.36
16	93.52	91.74	95.2	95.04	96.57	96.42	97.24	97.11	97.64	97.52
17	94.44	91.74	95.63	95.45	96.86	96.69	97.45	97.31	97.8	97.69

Table 7.10 illustrates the results of X-Linked-EDMD data set, the highest classification accuracy in the training data set (97.80%) and test data set (97.69%) is obtained using RBF SVM. This data set has only 17 genes selected. In case of the Caplain-3 data set, again RBF SVM has outperformed which is demonstrated in table 7.11. The maximum classification accuracy attained in the training data set (94.44%) and test data set (92.31%) using only 4 selected genes are very high. The Dysferlin data set has 8 genes selected and the results of using them are shown in table 7.12. The best classification accuracy is achieved in the training data set (92.57%) and test data set (92.56%) using RBF SVM. Table 7.13 shows the results of using different classification algorithms on FKRP dataset. The best classification accuracy is obtained in the training data set (95.44%) and test data set (95.37%) using RBF SVM. In case of NHSM data set, the outperformer classification algorithm RBF SVM has given the highest classification algorithm in the training data set (92.36%) and test data set (92.36%) using only the 19 selected genes. The results on this data set are shown in table 7.14.

Table 7.11: LOOCV Performance measures for Caplain-3 data set during different runs

Run	**Classification algorithms**									
	LDA		**QDA**		**KNN**		**Linear SVM**		**RBF SVM**	
	TR	**TS**	**TR**	**TS**	**TR**	**TS**	**TR**	**TS**	**TR**	**TS**
1	58.33	58.68	65.5	66.12	76.57	76.86	80.25	80.58	82.6	82.81
2	66.67	66.12	65.07	64.88	75.71	75.76	79.41	79.75	81.93	82.15
3	63.89	63.64	65.07	64.88	75.71	75.48	79.19	79.55	82.09	82.31
4	67.59	53.85	75	84.62	100	76.92	91.67	92.31	94.44	92.31

Table 7.12: LOOCV Performance measures for Dysferlin data set during different runs

Run	Classification algorithms									
	LDA		QDA		KNN		Linear SVM		RBF SVM	
	TR	TS	TR	TS	TR	TS	TR	TS	TR	TS
1	69.44	70.25	70.31	70.66	79.71	79.89	82.59	82.85	84.46	84.63
2	70.37	71.07	74.67	74.38	82.29	82.64	84.71	84.92	86.15	86.28
3	73.15	73.55	76.86	77.69	84.57	84.85	86.41	86.57	87.5	87.6
4	70.37	71.07	77.29	78.1	84.86	85.12	86.62	86.78	87.67	87.77
5	78.7	77.69	81.66	82.23	87.71	87.88	88.75	88.84	90.03	90.08
6	78.7	77.69	84.72	84.71	89.43	89.53	90.02	90.08	91.05	91.07
7	76.85	76.03	84.72	84.71	89.43	89.53	90.02	90.08	91.39	91.4
8	77.78	76.86	87.34	87.6	91.43	91.46	91.51	91.53	92.57	92.56

Table 7.13: LOOCV Performance measures for FKRP data set during different runs

Run	Classification algorithms									
	LDA		QDA		KNN		Linear SVM		RBF SVM	
	TR	TS	TR	TS	TR	TS	TR	TS	TR	TS
1	56.48	56.2	49.78	49.59	65.14	66.12	72.61	73.14	77.03	77.36
2	81.48	80.17	86.46	86.78	90.86	90.63	91.51	91.53	92.4	92.4
3	79.63	79.34	85.59	85.95	90.29	90.36	91.3	91.32	92.23	92.23
4	77.78	77.69	83.41	83.47	88.57	88.71	90.02	90.08	91.22	91.24
5	77.78	77.69	84.28	84.3	89.14	89.26	90.45	90.5	91.55	91.57
6	83.33	82.64	84.72	84.3	89.14	89.26	90.45	90.5	91.55	91.57
7	84.26	83.47	83.84	83.88	88.86	88.98	90.23	90.29	91.39	91.4
8	86.11	85.95	86.03	85.54	90	90.08	91.51	91.53	92.4	92.4
9	86.11	85.12	86.46	87.19	91.14	90.91	92.14	92.15	93.24	93.22
10	86.11	85.12	86.9	86.78	90.86	90.91	92.14	92.15	93.41	93.39
11	83.33	80.17	86.9	85.95	90.29	90.08	91.51	91.53	93.07	93.06
12	85.19	81.82	87.34	86.36	90.57	90.36	91.72	91.74	93.24	93.22
13	87.96	85.12	90.39	90.08	93.14	92.84	93.63	93.6	94.76	94.71
14	87.04	84.3	89.52	88.84	92.29	92.29	93.21	93.18	94.43	94.38
15	89.81	87.6	92.14	90.91	93.71	93.66	94.27	94.21	95.27	95.21
16	89.81	87.6	92.58	91.74	94.29	93.94	94.48	94.42	95.44	95.37

Table 7.14: LOOCV Performance measures for NHSM data set during different runs

Run	Classification algorithms									
	LDA		QDA		KNN		Linear SVM		RBF SVM	
	TR	TS	TR	TS	TR	TS	TR	TS	TR	TS
1	56.48	56.2	52.4	52.89	67.43	68.04	71.76	72.31	74.49	74.88
2	53.7	53.72	48.91	47.93	64	64.46	69	69.63	72.3	72.73
3	57.41	57.85	55.46	55.37	69.14	69.7	73.04	73.55	75.51	75.87
4	66.67	66.94	60.26	59.92	72.29	73	75.58	76.03	77.7	78.02
5	63.89	64.46	59.83	59.5	72	73	75.58	76.03	78.72	79.01
6	63.89	64.46	59.83	59.5	72	72.18	74.95	75.41	78.72	79.01
7	65.74	66.12	62.45	62.4	74	74.38	76.65	77.07	80.41	80.66
8	65.74	66.12	62.88	62.81	74.29	74.38	76.65	77.07	80.91	81.16
9	72.22	71.9	78.17	78.93	85.43	85.4	84.93	85.12	87.67	87.77
10	72.22	71.9	79.48	80.17	86.29	85.95	85.56	85.74	88.34	88.43
11	74.07	74.38	83.84	84.3	89.14	88.43	87.47	87.6	89.86	89.92
12	71.3	72.73	83.41	83.47	88.57	88.43	87.47	87.6	89.86	89.92
13	73.15	73.55	85.15	85.54	90	89.81	88.11	88.22	90.37	90.41
14	73.15	73.55	86.03	86.36	90.57	90.36	89.17	89.05	91.05	91.07
15	79.63	79.34	89.08	89.26	92.57	91.74	90.45	90.29	92.06	92.07
16	75	75.21	86.9	87.19	91.14	90.36	89.38	89.26	91.22	91.24
17	80.56	80.17	78.6	78.93	85.43	85.12	86.62	86.57	89.02	89.09
18	76.85	76.03	76.86	76.86	84	83.75	85.35	85.54	88.18	88.26
19	80.56	79.34	78.17	78.1	89.08	88.43	90	90.08	92.36	92.36

The other important highlights of the proposed integrated gene selection and classification model are listed. It includes only those predicted genes in the biomarker gene subsets whose expression is representative of the specific type of NMD. It reduces the misclassification rates by selecting only discriminating genes for each class of NMD. It can be effectively used by novice researchers as it does not require much domain knowledge. As the integrated model is classifier independent so it can be easily combined with any of the classification algorithms. We need not to specify the classification algorithm beforehand. But from the results, it is clear that the integration of RBF SVM with our gene selection method has outperformed all other integrations. It also provides flexibility for data division as the prior knowledge of gene expression profiles is not required. Our model has very simple computational steps and is very easy to implement.

7.4. Conclusions

The microarray data set of NMDs is cursed from high dimensionality as it contains tens of thousands of genes. The dimension of these data sets needs to be reduced in order to find out the compact subsets of biomarker genes that correctly classify these diseases. This will benefit the classifier by increasing classification performance, decreasing computational load and complexity. So in the present chapter, our motive is to search for the smallest gene subsets for every class which guarantee the accurate classification of diseases. We propose an integrated model for gene selection and multi-class classification of NMDs employing OVA approach. The proposed method is applied to the huge microarray data set of the NMD data set which contains 22,645 genes and 121 samples. The prominent conclusion of the present work is that we could identify those biomarker genes which are only needed to accurately classify the kinds of NMDs. Our method has significantly reduced the misclassification rates by using only those genes for classification. It has selected a minimum of 4 genes in one class and a maximum of 19 genes in another class. It is worth pointing out that by using only 1 gene per class, we could predict the kind of NMD of samples up to 94.59% and 94.55% in training and test datasets respectively under the LOOCV scheme. The integration of the proposed method of gene selection with RBF SVM has given highest classification accuracies amongst all.

The further investigation of these biomarker gene subsets is required as the classification accuracies obtained by the combination of these genes is very high. The medical relevance of these genes will be useful in discovering the drugs for the treatment of these diseases. It will be easier for the future researchers give better treatment options by visualizing the expression profiles of these genes.

The work mentioned in this chapter has been accepted for publication in International Journal of Computational Biology and Drug Design [Forthcoming issue].

Chapter 8

Conclusions and Future Work

In this chapter, the conclusions and the future scope of the proposed work are given. This chapter is structured as follows: Section 8.1 presents the conclusions. Section 8.2 gives the future work of the thesis.

8.1 Conclusions

For the last few years, it has been known that with the help of gene expression data it is possible to accurately diagnose any disease. The microarray technology has made it feasible to check the activity of gene expressions of a whole organism. This thesis presents a computational intelligent method for gene selection and classification of NMDs. The method selects only a few genes for the accurate classification.

In chapter 1 we begin with the introduction of computational intelligence methods and the use of these methods for gene selection and classification. This is followed by a detailed literature review on the use of individual and integrated knowledge based methods and computational intelligence methods for solving various challenges in bioinformatics. The chapter ends, presenting the motivation, plan of the thesis and its summary. From the literature review, it was found that the gene selection and classification of binary-class and multi-class datasets is a primary concern for the accurate and correct diagnosis of NMDs in medical field. This necessitates the development of an integrated method utilizing computational intelligence methods for gene selection and classification which selects only few genes from a large datasets and does the correct classification. As a result, the proposed work has been conceded out giving extraordinary attention to selection of only few or optimal genes which provides us with the highest accuracy.

Chapter 2 presents the basic concepts of gene selection and classification. It includes the biological background information with the problem statement of NMD classification, issues of disease classification, publicly available datasets, gene selection and its categories, various techniques of gene selection, classification algorithms and model validation techniques.

Chapter 3 provides an unsupervised approach for the diagnosis of facioscapulohumeral muscular dystrophy using cosine distance metric-hierarchical clustering algorithm and k-nearest neighbor based methodology where the former was used for feature selection and the latter was used for classification. The proposed method is evaluated on a dataset consisting of 50 samples and 33,297 genes. The genes in the dataset were ranked using Wilcoxon rank sum test. Followed by that, the clustering of genes and classification of dataset has been done. The experimental results shows that the integrated methods, i.e., k-means-LDA, k-means-QDA, k-means-KNN, euclidean distance metric-hierarchical clustering algorithm-LDA, euclidean distance metric-hierarchical clustering algorithm-QDA, euclidean distance metric-hierarchical clustering algorithm-KNN, cosine distance metric-hierarchical clustering algorithm-LDA and cosine distance metric-hierarchical clustering algorithm-QDA have not worked well for gene selection and clustering of NMD. The results were compared in terms of accuracy, sensitivity, specificity, positive predicted value and negative predicted value employing holdout validation technique and are shown in tables 3.1 to 3.3. The proposed method selects only few genes, i.e., 500 genes out of 33,297 genes and gives the classification accuracy of 87.39% which is much higher as compared to other reported integrated approaches. This is due to the fact that in hierarchical clustering method, there is no need to define the number of clusters beforehand. The cophenetic correlation coefficient using cosine distance metric is found to be better as compared to the euclidean distance metric. The hierarchical clustering algorithm with cosine distance metric is giving high performance because it takes into account the relative sizes rather than the absolute sizes of observations.

Chapter 4 deals with an integrated method for dimension reduction and classification applied to microarray data of NMDs employing entropy based feature selection technique and linear SVM for classification. The proposed integrated method was evaluated on two datasets, i.e., juvenile dermatomyositis and facioscapulohumeral muscular dystrophy containing 39 and 32 samples respectively. Both of the datasets contain a total of 22,645 genes. The experiment of the proposed method was run in MATLAB. Two filter techniques, namely t-test and entropy are used for gene selection and followed by that two classification algorithms linear SVM and KNN are deployed for classification. The results show that the integrated methods like t-test-linear SVM, entropy-linear SVM did not perform well, whereas t-test-KNN showed much better performance comparatively. But the best performance measures were given by the integrated method, i.e., entropy-KNN. Again, the performance measures were accuracy, sensitivity, specificity, positive predicted value and negative predicted value employing five-fold cross-validation technique and are depicted in tables 4.1 to 4.4. The proposed method selects 500 genes out of total 22,645 genes from both the datasets and has shown the accuracy of 92.31% in JDM and 96.88% in FSHD dataset.

Chapter 5 presents a novel intelligent integrated method of gene selection for facioscapulohumeral muscular dystrophy diagnosis deploying genetic algorithm and KNN. The proposed integrated method was evaluated on a dataset of facioscapulohumeral muscular dystrophy containing 33,297 genes and 50 samples. The genes are filtered and ranked using t-test. Genetic algorithm was employed to select the most discriminating genes where the fitness function was calculated using LDA, QDA and KNN one after the other. The experimentations were taking different numbers of genes every time. In the genetic algorithm, when the fitness function is evaluated using KNN, it gave the best performance measures. Other two integrations namely genetic algorithm-LDA and genetic algorithm-QDA did not perform much well comparatively. Again, the performance measures were accuracy, sensitivity, specificity, positive predicted value and negative predicted value and the model if validated using leave-one-out cross-validation techniques whose results are shown in table 5.2. The proposed integrated method, genetic algorithm-KNN gave 100% classification accuracy by selecting just 10 top most genes out of 33,297 genes. The model is validated using the leave-one-out cross-validation technique.

Chapter 6 presents a novel hybrid feature selection model employing Bhattacharyya coefficient, genetic algorithm and radial basis function based support vector machine for classification of NMDs. The proposed integrated method was evaluated on two multi-class datasets of NMDs. First dataset consisted of 22,645 genes, 5 classes and 72 numbers of samples, whereas second dataset consisted of 22,645 genes, 6 classes and 55 numbers of samples. The genes in both of the datasets are ranked and filtered using Bhattacharyya technique and then selected using genetic algorithm where the fitness function is calculated using LDA, QDA, KNN, linear SVM and RBF SVM. The proposed hybrid approach GA-RBF SVM performed best as compared to other implemented approaches. The hybridization of GA with LDA, QDA, and KNN has given satisfactory performance and linear SVM has also performed better. The performance measure, i.e., accuracy was calculated for all the hybridized methods, the model is validated using five-fold cross-validation technique and the results are shown in table 6.5. The hybridization of GA with RBF SVM has given 98.464%

and 98.48% in first and second data sets respectively, and it has selected only 100 genes out of 22,645 genes.

Chapter 7 presents a novel approach for the dissimilar gene selection and multi-class classification of NMDs by combining median matrix and radial basis function based support vector machine. The proposed novel intelligent method was evaluated on the dataset of 22,645 genes, 13 classes and 121 number of samples. Here the gene expression matrix is preprocessed in such a way that it creates a median matrix for the selection of few compact subsets of genes. After selection of gene subsets, the samples are classified using LDA, QDA, KNN, linear SVM and RBF SVM. Here also RBF SVM performed best as compared to other classifiers. The very few genes are selected for each class, i.e., 4 of class AQM, 10 for class ALS, 15 for class BMD, 14 for class FSHD, 15 for class JD, 12 for class DMD, 14 for class SPG4, 9 for class AD-EDMD, 17 for class X-Linked-EDMD, 4 for class Caplain-3, 8 for class Dysferlin, 16 for FKRP and 19 for NHSM out of 22,645 genes. The performance measure, i.e., classification accuracy is calculated in a different way by combining the genes. Here, in the first iteration, first selected gene is used for classification and if we do not get the satisfactory classification performance, then in the next iteration, we added next gene to the subset and so on till we get the satisfactory classification performance.

Chapter 8 discusses the conclusions of all the chapters and presents the future work of the thesis.

8.2 Future Works

The accurate diagnosis of NMDs employing computational intelligence methods provides a path for the future researchers to discover its evolution mechanism, prevention, cure and drug discovery for the treatment. Due to these computational intelligence methods, it will be possible for them to find out the interacting genes related to NMD development.

List of Publications

Published:

1. Divya Anand, Babita Pandey, Devendra K Pandey, A Novel Hybrid Feature Selection Model for Classification of Neuromuscular Dystrophies Using Bhattacharyya Coefficient, Genetic Algorithm and Radial Basis Function Based Support Vector Machine. Interdisciplinary Sciences: Computational Life Sciences, Vol. 10, No. 2, 2018, pp. 244-250, Springer Berlin Heidelberg, Germany, DOI: 10.1007/s12539-016-0183-6.

2. Divya Anand, Babita Pandey, Devendra K Pandey, Knowledge and Intelligent Computing Techniques in Bioinformatics. International Journal of Computational Biology and Drug Design, Vol. 9, No. 3, pp. 173-227, 2016, Inderscience Publishers, United Kingdom, DOI: 10.1504/IJCBDD.2016.078277.

3. Divya Anand, Babita Pandey, Devendra K Pandey, An Integrated Algorithm for Dimension Reduction and Classification Applied to Microarray Data of Neuromuscular Dystrophies. Indian Journal of Science and Technology, Vol. 9, No. 28, pp. 1-6, 2016, India, DOI: 10.17485/ijst/2016/v9i28/98378.

4. Divya Anand, Babita Pandey, Devendra K Pandey, Building an Intelligent Integrated Method of Gene Selection for Facioscapulohumeral Muscular Dystrophy Diagnosis. International Journal of Biomedical Engineering and Technology, Vol. 24, No. 3, pp. 285-296, 2017, Inderscience Publishers, United Kingdom, DOI: 10.1504/IJBET.2017.085144.

5. Divya Anand, Babita Pandey, Devendra K Pandey, Facioscapulohumeral Muscular Dystrophy Diagnosis using Hierarchical Clustering Algorithm and K-Nearest Neighbor based Methodology. International Journal of E-Health and Medical Communications, Vol. 8, No. 2, pp. 33-46, 2017, IGI Global Publishing, United States, DOI: 10.4018/IJEHMC.2017040103.

Communicated:

1. Divya, Babita Pandey, Devendra K Pandey, A novel approach for dissimilar gene selection and multi-class classification of neuromuscular disorders: Combining median matrix and radial basis function based support vector machine. International Journal of Computational Biology and Drug Design, Inderscience Publishers [Forthcoming Issue].

References

[1] G. Valentini, R. Tagliaferri, and F. Masulli, “Computational Intelligence and Machine Learning in Bioinformatics,” *Artif. Intell. Med.*, vol. 45, no. 2–3, pp. 91–96, 2009.

[2] Z. Ghosh and B. Mallick, *Bioinformatics Principles and Applications*. Oxford University Press, 2008.

[3] R. Akerkar and P. Sajja, “Knowledge-Based Systems,” *Knowledge-Based Syst.*, vol. 23, no. 5, pp. 1–114, 2010.

[4] S. Montani, G. Leonardi, S. Ghignone, and L. Lanfranco, “Flexible, efficient and interactive retrieval for supporting in-silico studies of endobacteria,” *Proc. - Int. Conf. Tools with Artif. Intell. ICTAI*, pp. 17–24, 2011.

[5] K. J. Cios, H. Mamitsuka, T. Nagashima, and R. Tadeusiewicz, “Computational intelligence in solving bioinformatics problems,” *Artif. Intell. Med.*, vol. 35, no. 1–2, pp. 1–8, 2005.

[6] P. Larranaga *et al.*, “Machine learning in bioinformatics,” *Brief. Bioinform.*, vol. 7, no. 1, pp. 86–112, 2006.

[7] G. D. Stormo, T. D. Schneider, L. Gold, and A. Ehrenfeucht, “Use of the ‘Perceptron’ algorithm to distinguish transational initiation sites in E. coli,” *Nucleic acid Res.*, vol. 10, no. 9, pp. 2297–3011, 1982.

[8] R. M. Jr, “Knowledge based systems and neural networks,” *Environ. Sci.*, vol. III, 2005.

[9] M. Lebowitz, “Memory-based parsing,” *Artif. Intell.*, vol. 21, no. 4, pp. 363–404, 1983.

[10] A. Aamodt and E. Plaza, “Case-Based Reasoning,” *Artif. Intell. Commun.*, vol. 7, no. 1, pp. 39–59, 1994.

[11] B. Pandey and R. B. Mishra, “Knowledge and intelligent computing system in medicine,” *Comput. Biol. Med.*, vol. 39, no. 3, pp. 215–230, 2009.

[12] L. Fu, “An Expert Network For DNA Sequence Analysis - IEEE Intelligent Systems” *IEEE Intell. Syst.*, 1999.

[13] D. Frias, F. Vidd, and J. C. M. Cascardo, “Finding gene promoters in the genome of the fungus crlnlpellls neural networks pernlclosa using feed-forward,” in *IEEE Workshop on Machine Learning for Signal Processing*, 2004, pp. 423–432.

[14] C.-C. Liu *et al.*, “Genome-wide identification of specific oligonucleotides using artificial neural network and computational genomic analysis.,” *BMC Bioinformatics*, vol. 8, p. 164, 2007.

[15] F. E. Ahmed, “Artificial neural networks for diagnosis and survival prediction in colon cancer.,” *Mol. Cancer*, vol. 4, p. 29, 2005.

[16] A. H. Chen and C. H. Lin, “A novel support vector sampling technique to improve classification accuracy and to identify key genes of leukaemia and prostate cancers,” *Expert Syst. Appl.*, vol. 38, no. 4, pp. 3209–3219, 2011.

[17] M. K. Gupta, K. Agarwal, N. Prakash, D. B. Singh, and K. Misra, “Prediction of miRNA in HIV-1 genome and its targets through artificial neural network: a bioinformatics approach,” *Netw. Model. Anal. Heal. Informatics Bioinforma.*, vol. 1, no. 4, pp. 141–151, 2012.

[18] T. J. S.- Ning Qian, “Predicting the secondary structure of globular proteins using neural network models,” *J. Mol. Biol.*, 1988.

[19] C. H. Wu, G. M. Whitson, C.-T. Hsiao, and C.-F. Huang, “Classification artificial neural systems for genome research,” in *Proceedings of 1992 ACM/IEEE Conference on Supercomputing*, 1992.

[20] X. Zhang, “A hybrid algorithm for determining protein structure,” *Ieee Expert Intell. Syst. Their Appl.*, 1994.

[21] C. Wu, M. Berry, S. Shivakumar, and J. Mclarity, “Neural Networks for Full-Scale Protein Sequence Classification: Sequence Encoding with Singular Value Decomposition,” *Mach. Learn.*, vol. 21, pp. 177–193, 1995.

[22] Y.-D. Cai and K.-C. Chou, “Artificial neural network model for predicting HIV protease cleavage sites in protein,” *Adv. Eng. Softw.*, vol. 29, no. 2, pp. 119–128, 1998.

[23] P. Baldi and G. Pollastri, “Strategy for Protein Analysis,” *IEEE Intell. Syst.*, 2002.

[24] H. Zhu, I. Yoshihara, and K. Yamamori, “A multimodal neural network with single-state predictions for protein secondary structure,” *Artif. Life*, pp. 168–173, 2004.

[25] A. Ceroni, P. Frasconi, and G. Pollastri, “Learning protein secondary structure from sequential and relational data,” *Neural Networks*, vol. 18, no. 8, pp. 1029–1039, 2005.

[26] J. Wang and J.-P. Li, “Protein secondary structure prediction based on bp neural network and quasi-newton algorithm,” in *International Conference on Apperceiving Computing and Intelligence Analysis, 2008. ICACIA 2008.*, 2008.

[27] R. Kakumani and V. Devabhaktuni, “A Two-Stage Neural Network Based Technique for Protein Secondary Structure Prediction,” in *30th Annual International IEEE EMBS Conference*, 2008.

[28] M. Nielsen and O. Lund, “NN-align. An artificial neural network-based alignment algorithm for MHC class II peptide binding prediction.,” *BMC Bioinformatics*, vol. 10, p. 296, 2009.

[29] W.-Z. Lin and X. Xiao, "Using grey neural network to predict protein primary structure," in *Proceedings of International Conference on Information Engineering and Computer Science,* 2009.

[30] N. P. Bidargaddi and J. Chetty, Madhu Kamruzzaman, "Combining segmentalsemi-Markovmodelswithneuralnetworksforprotein secondary structureprediction," *Neurocomputing*, vol. 72, pp. 3943–3950, 2009.

[31] H. Mathkour and M. Ahmad, "An integrated approach for protein structure prediction using artificial neural network," in *2010 Second International Conference on Computer Engineering and Applications*, 2010.

[32] G. Kim, Y. Kim, H. Lim, and H. Kim, "An MLP-based feature subset selection for HIV-1 protease cleavage site analysis," *Artif. Intell. Med.*, vol. 48, pp. 83–89, 2010.

[33] R. Priyadarshini, N. Dash, and S. Rout, "A Novel Approach for Protein Structure Prediction using Back Propagation Neural Network," *Int. J. Comput. Sci ence Technol.*, vol. 3, no. 2, pp. 600–603, 2012.

[34] J. P. Florido, H. Pomares, I. Rojas, A. Guillen, F. M. Ortuno, and J. M. Urquiza, "An effective, practical and low computational cost framework for the integration of heterogeneous data to predict functional associations between proteins by means of artificial neural networks," *Neurocomputing*, vol. 121, pp. 64–78, 2013.

[35] C. Deng, P. Zhang, A. Wang, B. J. Trummer, and D. Wang, "Normalization of cDNA Microarray Data By Using Neural Networks," in *Proceedings of International Joint Conference on Neural Networks, IEEE,* 2002, pp. 290–295.

[36] D. P. Berrar, C. S. Downes, and W. Dubitzky, "Multiclass cancer classification using gene expression profiling and probabilistic neural networks.," *Pac. Symp. Biocomput.*, vol. 16, pp. 5–16, 2003.

[37] B. Rost and C. Sander, "Prediction of protein secondary structure at better than 70% accuracy.," *Journal of molecular biology,* vol. 232, no. 2. pp. 584–599, 1993.

[38] B. Rost, C. Sander, and R. Schneider, "Evolution and neural networks/spl minus/protein secondary structure prediction above 71% accuracy - System Sciences, 1994. Vol.V: Biotechnology Computing, Proceedings of the Twenty-Seventh Hawaii In," pp. 385–394, 1994.

[39] J. M. Chandonia and M. Karplus, "Neural networks for secondary structure and structural class predictions," *Protein Sci.*, vol. 4, no. 2, pp. 275–285, 1995.

[40] L. Jian-wei, C. Guang-hui, H. Li, L. Yuan, and X. Luo, "Prediction of Protein Secondary Structure Using Multilayer Feed-forward Neural Networks," in *25th Chinese Control and Decision Conference (CCDC)*, pp. 1346–1351, 2013.

[41] H. B. Kazemian, S. A. Yusuf, and K. White, "Signal peptide discrimination and cleavage site identification using SVM and NN," *Comput. Biol. Med.*, vol. 45, no. 1,

pp. 98–110, 2014.

[42] D. L. Tong and A. C. Schierz, “Hybrid genetic algorithm-neural network: Feature extraction for unpreprocessed microarray data,” *Artif. Intell. Med.*, vol. 53, no. 1, pp. 47–56, 2011.

[43] L. De Campos Teixeira Gomes, F. J. Von Zuben, and P. Moscato, “A proposal for direct-ordering gene expression data by self-organising maps,” *Appl. Soft Comput. J.*, vol. 5, no. 1, pp. 11–21, 2004.

[44] S. S. Sahu, G. Panda, and R. Barik, “A Hybrid Method of Feature Extraction for Tumor Classification Using Microarray Gene Expression Data,” vol. 1, no. 1, 2011.

[45] J. Chen and N. Chaudhari, “Capturing long-term dependencies for protein secondary structure prediction,” *Adv. Neural Networks-ISNN 2004*, pp. 1–6, 2004.

[46] Y. Wang, Y. Lin, M. Shu, R. Wang, Y. Hu, and Z. Lin, “Proteasomal cleavage site prediction of protein antigen using BP neural network based on a new set of amino acid descriptor,” *J. Mol. Model*, vol. 19, pp. 3045–3052, 2013.

[47] A. Sharma and K. K. Paliwal, “Cancer classification by gradient LDA technique using microarray gene expression data,” *Data Knowl. Eng.*, vol. 66, no. 2, pp. 338–347, 2008.

[48] S. Knott, S. Mostafavi, and P. Mousavi, “A neural network based modeling and validation approach for identifying gene regulatory networks,” *Neurocomputing*, vol. 73, no. 13–15, pp. 2419–2429, 2010.

[49] M. V. Thalatam, P. V. Rao, K. Varma, N. Murty, and A. llam Apparao, “Prediction of Protein Secondary Structure using Artificial Neural Network,” *Int. J. Comput. Sci. Eng.*, vol. 2, no. 5, pp. 1615–1621, 2010.

[50] Z. R. Yang, J. Dry, R. Thomson, and T. Charles Hodgman, “A bio-basis function neural network for protein peptide cleavage activity characterisation,” *Neural Networks*, vol. 19, no. 4, pp. 401–407, 2006.

[51] Y. Zhang, C.-H. Chu, Y. Chen, H. Zha, and X. Ji, “Splice site prediction using support vector machines with a bayes kernel,” *Expert Syst. Appl.*, vol. 30, no. 1, pp. 73–81, 2006.

[52] W. Fei and C. Lusheng, “Detecting DNA-binding Domain From Sequence and Secondary Structure Information Using Kernel-based Technique,” in *Proceedings of 2008 3rd International Conference on Intelligent System and Knowledge Engineering*, 2008, pp. 200–204.

[53] N. Beerenwinkel *et al.*, “Geno2pheno,” no. December, pp. 35–41, 2001.

[54] I. Guyon, J. Wetson, and S. Barnhill, “Gene Selection for Cancer Classification using Support Vector Machines,” *Mach. Learn.*, vol. 46, pp. 389–422, 2002.

[55] J. Zhang and H.-W. Deng, “Gene selection for classification of microarray data based on the Bayes error.,” *BMC Bioinformatics*, vol. 8, no. 1, p. 370, 2007.

[56] S. Zheng and W. Liu, “An experimental comparison of gene selection by Lasso and Dantzig selector for cancer classification,” *Comput. Biol. Med.*, vol. 41, no. 11, pp. 1033–1040, 2011.

[57] C. Arunkumar and S. Ramakrishnan, “Two Step Feature Extraction Method for Microarray Cancer Data using Support Vector Machines,” *Int. J. Comput. Appl.*, vol. 85, no. 8, pp. 34–42, 2014.

[58] J. M. Urquiza, I. Rojas, L. J. Pomares, H. Herrera, J. Ortega, and A. Prieto, “Method for prediction of protein–protein interactions in yeast using genomics/proteomics information and feature selection,” *Neurocomputing*, vol. 74, pp. 2683–2690, 2011.

[59] X.-Q. Zeng, G.-Z. Li, J. Y. Yang, M. Q. Yang, and G.-F. Wu, “Dimension reduction with redundant gene elimination for tumor classification.,” *BMC Bioinformatics*, vol. 9 Suppl 6, no. Suppl 6, p. S8, 2008.

[60] M. N. Nguyen and J. C. Rajapakse, “Prediction of protein secondary structure with two-stage multi-class SVMs,” *Int. J. Data Min. Bioinform.*, vol. 1, no. 3, pp. 248–269, 2007.

[61] R. Z. Aram and N. M. Charkari, “A two-layer classification framework for protein fold recognition.,” *J. Theor. Biol.*, vol. 365, pp. 32–39, 2014.

[62] S. Hua and Z. Sun, “A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach.,” *J. Mol. Biol.*, vol. 308, no. 2, pp. 397–407, 2001.

[63] H.-J. Hu, Y. Pan, R. Harrison, and P. C. Tai, “Improved Protein Secondary Structure Prediction Using Support Vector Machine With a New Encoding Scheme and an Advanced Tertiary Classifier,” *IEEE Trans. Nanobioscience*, vol. 3, no. 4, pp. 30303–34110, 2004.

[64] L.-H. Wang, J. Liu, Y.-F. Li, and H.-B. Zhou, “Predicting protein secondary structure by a support vector machine based on a new coding scheme.,” *Genome Inform.*, vol. 15, no. 2, pp. 181–90, 2004.

[65] R. Bi, Y. Zhou, F. Lu, and W. Wang, “Predicting Gene Ontology functions based on support vector machines and statistical significance estimation,” *Neurocomputing*, vol. 70, no. 4–6, pp. 718–725, 2007.

[66] L. Wang, F. Chu, and W. Xie, “Expressions of Very Few Genes,” vol. 4, no. 1, pp. 40–53, 2007.

[67] Z. J. Lee, “An integrated algorithm for gene selection and classification applied to microarray data of ovarian cancer,” *Artif. Intell. Med.*, vol. 42, no. 1, pp. 81–93, 2008.

[68] W. Qu, B. Yang, Z. Ying, and H. Sui, “Predicting protein secondary structure using a mixed-modal SVM method in a compound pyramid model,” *Knowledge-Based Syst.*, vol. 24, pp. 304–313, 2011.

[69] A. Korfiati, K. Theofilatos, D. Kleftogiannis, C. Alexakos, S. Likothanassis, and S. Mavroudi, “Predicting human miRNA target genes using a novel computational intelligent framework,” *Inf. Sci. (Ny).*, vol. 294, no. October, pp. 576–585, 2015.

[70] J. Ding, S. Zhou, and J. Guan, “miRFam: an effective automatic miRNA classification method based on n-grams and a multiclass SVM.,” *BMC Bioinformatics*, vol. 12, no. 1, p. 216, 2011.

[71] C. K. Chen, “The classification of cancer stage microarray data,” *Comput. Methods Programs Biomed.*, vol. 108, no. 3, pp. 1070–1077, 2012.

[72] M. N. Nguyen and J. C. Rajapakse, “Multi-Class Support Vector Machines for Protein Secondary Structure Prediction,” *Genome Informatics*, vol. 14, pp. 218–227, 2003.

[73] Y. D. Cai, Y. X. Li, and K. C. Chou, “Classification and prediction of ??-turn types by neural network,” *Adv. Eng. Softw.*, vol. 30, no. 5, pp. 347–352, 1999.

[74] F. Azuaje, “Making genome expression data meaningful: Prediction and discovery of classes of cancer through a connectionist learning approach,” *Proc. - IEEE Int. Symp. Bio-Informatics Biomed. Eng. BIBE 2000*, pp. 208–213, 2000.

[75] I. Yoshihara, “Feature Extraction from Genome Sequence Using Multi-Modal Neural Network,” vol. 422, pp. 420–422, 2001.

[76] X. Xu and A. Zhang, “Virtual Gene: A Gene Selection Algorithm for Sample Classification on Microarray Datasets.,” *2005 Int. Work. Bioinforma. Res. Appl.*, pp. 1038–1045, 2005.

[77] K. Yendrapalli, R. Basnet, S. Mukkamala, and A. H. Sung, “Gene Selection for Tumor Classification Using Microarray Gene Expression Data,” vol. I, pp. 4–9, 2007.

[78] K. Nakayama, A. Hirano, and K. -i. Fukumura, “On generalization of multilayer neural network applied to predicting protein secondary structure,” *2004 IEEE Int. Jt. Conf. Neural Networks (IEEE Cat. No.04CH37541)*, vol. 2, no. 4, pp. 1209–1213, 2004.

[79] B. Tang, X. Wang, and X. Wang, “Protein Secondary Structure Prediction using Large Margin Methods,” in *2009 Eigth IEEE/ACIS International Conference on Computer and Information Science*, 2009.

[80] W. Qu, H. Sui, B. Yang, and W. Qian, “Improving protein secondary structure prediction using a multi-modal BPmethod,” *Comput. Biol. Med.*, vol. 41, pp. 946–959, 2011.

[81] E. Abbasi, M. Ghatee, and M. E. Shiri, “FRAN and RBF-PSO as two components of a hyper framework to recognize protein folds,” *Comput. Biol. Med.*, vol. 43, no. 9, pp.

1182–1191, 2013.

[82] H.-Q. Wang, H.-S. Wong, H. Zhu, and T. T. C. Yip, "A neural network-based biomarker association information extraction approach for cancer classification.," *J. Biomed. Inform.*, vol. 42, no. 4, pp. 654–66, 2009.

[83] J. B. La, ˙ Zewicz, M. Kasprzak, and W. Kuroczycki, "Hybrid Genetic Algorithm for DNA Sequencing with Errors *," *J. Heuristics*, vol. 8, pp. 495–502, 2002.

[84] M. Kaya, "MOGAMOD: Multi-objective genetic algorithm for motif discovery," *Expert Syst. Appl.*, vol. 36, no. 2 PART 1, pp. 1039–1047, 2009.

[85] S. Vijayvargiya and P. Shukla, "A niched Pareto genetic algorithm for finding variable length regulatory motifs in DNA sequences," *3 Biotech*, vol. 2, no. 2, pp. 141–148, 2012.

[86] F. Ortuno, J. P. Florido, J. M. Urquiza, H. Pomares, A. Prieto, and I. Rojas, "Optimization of multiple sequence alignment methodologies using a multiobjective evolutionary algorithm based on NSGA-II," *2012 IEEE Congr. Evol. Comput.*, pp. 318–325, 2012.

[87] V. Di Gesú, R. Giancarlo, G. Lo Bosco, A. Raimondi, and D. Scaturro, "GenClust: a genetic algorithm for clustering gene expression data.," *BMC Bioinformatics*, vol. 6, p. 289, 2005.

[88] J. Thompson and S. Gopal, "Genetic algorithm learning as a robust approach to RNA editing site prediction.," *BMC Bioinformatics*, vol. 7, p. 145, 2006.

[89] C. C. To and J. Vohradsky, "A parallel genetic algorithm for single class pattern classification and its application for gene expression profiling in Streptomyces coelicolor.," *BMC Genomics*, vol. 8, p. 49, 2007.

[90] M. Perez, D. M. Rubin, T. Marwala, L. E. Scott, and W. Stevens, "A Population-Based Incremental Learning approach to microarray gene expression feature selection," *2010 IEEE 26-th Conv. Electr. Electron. Eng. Isr.*, pp. 000010–000014, 2010.

[91] S. Nemati, M. E. Basiri, N. Ghasem-Aghaee, and M. H. Aghdam, "A novel ACO-GA hybrid algorithm for feature selection in protein function prediction," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 12086–12094, 2009.

[92] S.-C. Su, C.-J. Lin, and C.-K. Ting, "An effective hybrid of hill climbing and genetic algorithm for 2D triangular protein structure prediction," *Proteome Sci.*, vol. 9, no. 1, 2011.

[93] F. L. Custodio, H. J. C. Barbosa, and L. E. Dardenne, "A multiple minima genetic algorithm for protein structure prediction," *Appl. Soft Comput.*, vol. 15, pp. 88–99, 2014.

[94] X. R. Jiang and P. Grünwald, "Microarray gene expression data association rules

mining based on BSC-tree and FIS-tree," *Data \& Knowl. Eng.*, vol. 53, pp. 3–29, 2005.

[95] R. Priscilla and S. Swamynathan, "A semi-supervised hierarchical approach: Two-dimensional clustering of microarray gene expression data," *Front. Comput. Sci.*, vol. 7, no. 2, pp. 204–213, 2013.

[96] W. Chmielnicki and K. Staçpor, "A hybrid discriminative/generative approach to protein fold recognition," *Neurocomputing*, vol. 75, pp. 194–198, 2012.

[97] P. C. H. Ma and K. C. C. Chan, "Inferring gene regulatory networks from expression data by discovering fuzzy dependency relationship," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 2, pp. 455–465, 2008.

[98] P. C. H. Ma and K. C. C. Chan, "Incremental fuzzy mining of gene expression data for gene function prediction," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 5, pp. 1246–1252, 2011.

[99] P. Maji and S. Paul, "Rough-Fuzzy Clustering for Grouping Functionally Similar Genes from Microarray Data," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 10, no. 2, pp. 1–1, 2012.

[100] D. L. González-Álvarez, M. A. Vega-Rodríguez, J. A. Gómez-Pulido, and J. M. Sánchez-Pérez, "Comparing multiobjective swarm intelligence metaheuristics for DNA motif discovery," *Eng. Appl. Artif. Intell.*, vol. 26, no. 1, pp. 314–326, 2013.

[101] S. Santander-Jiménez and M. A. Vega-Rodríguez, "Applying a multiobjective metaheuristic inspired by honey bees to phylogenetic inference," *BioSystems*, vol. 114, no. 1, pp. 39–55, 2013.

[102] M. S. Mohamad, S. Omatu, S. Deris, and M. Yoshioka, "Particle swarm optimization with a modified sigmoid function for gene selection from gene expression data," *Artif. Life Robot.*, vol. 15, no. 1, pp. 21–24, 2010.

[103] B. Wei, Q. K. Peng, Q. W. Zhang, and C. Y. Li, "Identification of a combination of SNPs associated with Graves' disease using swarm intelligence," *Sci. China Life Sci.*, vol. 54, no. 2, pp. 139–145, 2011.

[104] B. Li, Y. Li, and L. Gong, "Protein secondary structure optimization using an improved artificial bee colony algorithm based on AB off-lattice model," *Eng. Appl. Artif. Intell.*, vol. 27, pp. 70–79, 2014.

[105] S. Dixon and X. Yu, "Bioinformatics Data Mining Using Artificial Immune Systems and Neural Networks," *Electr. Eng.*, pp. 440–445, 2010.

[106] K. Anandakumar and M. Punithavalli, "Efficient Cancer Classification using Fast Adaptive Neuro-Fuzzy Inference System (FANFIS) based on Statistical Techniques," *IJACSA) Int. J. Adv. Comput. Sci. Appl. Spec. Issue Artif. Intell.*, pp. 132–137, 2011.

[107] D. Neagu and V. Palade, "A neuro-fuzzy approach for functional genomics data interpretation and analysis," *Neural Comput. Appl.*, vol. 12, no. 3–4, pp. 153–159, 2003.

[108] B. Yao and S. Li, "ANMM4CBR: a case-based reasoning method for gene expression data classification.," *Algorithms Mol Biol*, vol. 5, p. 14, 2010.

[109] A. Nikolova, V. Mladenov, Tsenov, and Georgi, "Performance comparison of techniques for DNA sequence prediction using neural network," in *4th International Symposium on Communications, Control and Signal Processing, ISCCSP 2010, Limassol*, 2010.

[110] C. Peterson and M. Ringnér, "Analyzing tumor gene expression profiles," *Artif. Intell. Med.*, vol. 28, no. 1, pp. 59–74, 2003.

[111] S. I. Ao and M. K. Ng, "Gene expression time series modeling with principal component and neural network," *Soft Comput.*, vol. 10, no. 4, pp. 351–358, 2006.

[112] L. Ziaei, A. R. Mehri, and M. Salehi, "Application of Artificial Neural Networks in Cancer Classification and Diagnosis Prediction of a Subtype of Lymphoma Based on Gene Expression Profile," *J. Res. Med. Sci.*, vol. 11, no. 1, 2006.

[113] K.-H. Chen *et al.*, "Gene selection for cancer identification: a decision tree model empowered by particle swarm optimization algorithm.," *BMC Bioinformatics*, vol. 15, no. 1, p. 49, 2014.

[114] Y.-C. Chen, W.-C. Ke, and H.-W. Chiu, "Risk classification of cancer survival using ANN with gene expression data from multiple laboratories," *Comput. Biol. Med.*, vol. 48, pp. 1–7, 2014.

[115] S. Kar, K. Das Sharma, and M. Maitra, "Gene selection from microarray gene expression data for classification of cancer subgroups employing PSO and adaptive K-nearest neighborhood technique," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 612–627, 2015.

[116] I. Turkoglu and E. D. Kaymaz, "A hybrid method based on artificial immune system and k-NN algorithm for better prediction of protein cellular localization sites," *Appl. Soft Comput. J.*, vol. 9, no. 2, pp. 497–502, 2009.

[117] E. Jacob, K. N. R. Nair, and R. Sasikumar, "A fuzzy-driven genetic algorithm for sequence segmentation applied to genomic sequences," *Appl. Soft Comput. J.*, vol. 9, no. 2, pp. 488–496, 2009.

[118] G. Schaefer and T. Nakashima, "Data mining of gene expression data by fuzzy and hybrid fuzzy methods," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 1, pp. 23–29, 2010.

[119] Z. Wang and V. Palade, "Building interpretable fuzzy models for high dimensional data analysis in cancer diagnosis," *BMC Genomics*, vol. 12, no. Suppl 2, p. S5, 2011.

[120] A. Zibakhsh and M. S. Abadeh, "Gene selection for cancer tumor detection using a novel memetic algorithm with a multi-view fitness function," *Eng. Appl. Artif. Intell.*, vol. 26, no. 4, pp. 1274–1281, 2013.

[121] E. G. Mansoori, M. J. Zolghadri, and S. D. Katebi, "Protein Superfamily Classification Using Fuzzy Rule-Based Classifier," *IEEE Trans. Nanobioscience*, vol. 8, no. 1, pp. 92–99, 2009.

[122] N. Kasabov, S. Pang, K. Engineering, P. Bag, and N. Zealand, "Transductive support vector machines and applications in biotnfomatics for promoter recognition," *Signal Processing*, pp. 1–6, 2003.

[123] K. Simek *et al.*, "Using SVD and SVM methods for selection, classification, clustering and modeling of DNA microarray data," *Eng. Appl. Artif. Intell.*, vol. 17, no. 4, pp. 417–427, 2004.

[124] C.-H. Zheng, Y.-W. Chong, and H.-Q. Wang, "Gene selection using independent variable group analysis for tumor classification," *Neural Comput. Appl.*, vol. 20, no. 2, pp. 161–170, 2011.

[125] S. Bose and C. Das, "A Novel Attribute Clustering Algorithm for Extraction of Discriminative Features to Classify Samples from Microarray Gene Expression Data," vol. 1, no. 4, pp. 148–153, 2013.

[126] J. He, H.-J. Hu, R. Harrison, P. C. Tai, and Y. Pan, "Rule generation for protein secondary structure prediction with support vector machines and decision tree.," *IEEE Trans. Nanobioscience*, vol. 5, no. 1, pp. 46–53, 2006.

[127] D. Shanthi, G. Sahoo, and N. Saravanan, "Input Feature Selection using Hybrid Neuro-Genetic Approach in the Diagnosis of Stroke Disease," *J. Comput. Sci.*, vol. 8, no. 12, 2008.

[128] A. H. Chen and J. Hsu, "Exploring novel algorithms for the prediction of cancer classification," no. 701, pp. 378–383, 2010.

[129] A. El Akadi, A. Amine, A. El Ouardighi, and D. Aboutajdine, "A two-stage gene selection scheme utilizing MRMR filter and GA wrapper," *Knowl. Inf. Syst.*, vol. 26, no. 3, pp. 487–500, 2011.

[130] R. Otwani, S. Ramrakhiani, and R. Rajpal, "Neural Network based Protein Structure Prediction," *IEEE*, pp. 408–412, 2003.

[131] L. Li *et al.*, "Data mining techniques for cancer detection using serum proteomic profiling," *Artif Intell Med*, vol. 32, no. 2, pp. 71–83, 2004.

[132] A. B. Reyaz-Ahmed, "Protein Secondary Structure Prediction Using Support Vector Machines, Neural Networks and Genetic Algorithms," Georgia State University, 2007.

[133] L. Nanni and A. Lumini, "A genetic approach for building different alphabets for

peptide and protein classification.," *BMC Bioinformatics*, vol. 9, no. 1, p. 45, 2008.

[134] Z. Li, X. Zhou, Z. Dai, and X. Zou, "Classification of G-protein coupled receptors based on support vector machine with maximum relevance minimum redundancy and genetic algorithm.," *BMC Bioinformatics*, vol. 11, p. 325, 2010.

[135] M. D. Ritchie, B. C. White, J. S. Parker, L. W. Hahn, and J. H. Moore, "Optimization of neural network architecture using genetic programming improves detection and modeling of gene-gene interactions in studies of human diseases.," *BMC Bioinformatics*, vol. 4, p. 28, 2003.

[136] N. Noman, L. Palafox, and H. Iba, "Reconstruction of Gene Regulatory Networks from Gene Expression Data Using Decoupled Recurrent Neural Network Model," in *Natural Computing and Beyond*, pp. 93–103, 2013.

[137] I. Maglogiannis, E. Zafiropoulos, and I. Anagnostopoulos, "An intelligent system for automated breast cancer diagnosis and prognosis using SVM based classifiers," *Appl. Intell.*, vol. 30, no. 1, pp. 24–36, 2009.

[138] Y. Lu, S. Lu, F. Fotouhi, Y. Deng, and S. J. Brown, "Incremental genetic K-means algorithm and its application in gene expression data analysis.," *BMC Bioinformatics*, vol. 5, p. 172, 2004.

[139] S. C. Shah and A. Kusiak, "Data mining and genetic algorithm based gene/SNP selection," *Artif. Intell. Med.*, vol. 31, no. 3, pp. 183–196, 2004.

[140] F. H. F. Han and N. R. N. Rao, "Mining Co-regulated Genes Using Association Rules Combined with Hash-tree and Genetic Algorithms," *2007 Int. Conf. Commun. Circuits Syst.*, pp. 858–862, 2007.

[141] F.-X. Wu, "Genetic weighted k-means algorithm for clustering large-scale gene expression data.," *BMC Bioinformatics*, vol. 9 Suppl 6, no. 6, p. S12, 2008.

[142] S. H. Aljahdali and M. E. El-Telbany, "Bio-inspired machine learning in microarray gene selection and cancer classification," *Signal Process. Inf. Technol. (ISSPIT), 2009 IEEE Int. Symp.*, pp. 339–343, 2009.

[143] S. R. Jangam and N. Chakraborti, "A novel method for alignment of two nucleic acid sequences using ant colony optimization and genetic algorithms," *Appl. Soft Comput.*, vol. 7, no. 3, pp. 1121–1130, 2007.

[144] J. F. De Paz, J. Bajo, V. Vera, and J. M. Corchado, "MicroCBR: A case-based reasoning architecture for the classification of microarray data," *Appl. Soft Comput. J.*, vol. 11, no. 8, pp. 4496–4507, 2011.

[145] K. Anekboon, C. Lursinsap, S. Phimoltares, S. Fucharoen, and S. Tongsima, "Extracting predictive SNPs in Crohn's disease using a vacillating genetic algorithm and a neural classifier in case-control association studies," *Comput. Biol. Med.*, vol. 44, no. 1, pp. 57–65, 2014.

[146] S. H. Doong and C. Y. Yeh, “Secondary Structure Prediction Using SVM and Clustering,” *Fourth Int. Conf. Hybrid Intell. Syst.*, pp. 297–302, 2004.

[147] L. Nanni and A. Lumini, “An ensemble of support vector machines for predicting virulent proteins,” *Expert Syst. Appl.*, vol. 36, no. 4, pp. 7458–7462, 2009.

[148] B. Liu, Q. Cui, T. Jiang, and S. Ma, “A combinational feature selection and ensemble neural network method for classification of gene expression data.,” *BMC Bioinformatics*, vol. 5, p. 136, 2004.

[149] F. Napolitano, G. Raiconi, R. Tagliaferri, A. Ciaramella, A. Staiano, and G. Miele, “Clustering and visualization approaches for human cell cycle gene expression data analysis,” *Int. J. Approx. Reason.*, vol. 47, no. 1, pp. 70–84, 2008.

[150] S. Karimi and M. Farrokhnia, “Leukemia and small round blue-cell tumor cancer detection using microarray gene expression data set: Combining data dimension reduction and variable selection technique,” *Chemom. Intell. Lab. Syst.*, vol. 139, pp. 6–14, 2014.

[151] M. Seera and C. P. Lim, “A hybrid intelligent system for medical data classification,” *Expert Syst. Appl.*, vol. 41, no. 5, pp. 2239–2249, 2014.

[152] F. Rahimov *et al.*, “Transcriptional profiling in facioscapulohumeral muscular dystrophy to identify candidate biomarkers.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 109, no. 40, pp. 16234–9, 2012.

[153] M. Bakay *et al.*, “Nuclear envelope dystrophies show a transcriptional fingerprint suggesting disruption of Rb-MyoD pathways in muscle regeneration,” *Brain*, vol. 129, no. 4, pp. 996–1013, 2006.

[154] R. A. FISHER, “The use of multiple measurements in taxonomic problems,” *Ann. Eugen.*, vol. 7, no. 2, pp. 179–188, 1936.

[155] M. L. Zhang and Z. H. Zhou, “ML-KNN: A lazy learning approach to multi-label learning,” *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, 2007.

[156] N. S. Altman, “An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression,” *Am. Stat.*, vol. 46, no. 3, pp. 175–185, 1992.

[157] E. Boser, N. Vapnik, I. M. Guyon, and T. B. Laboratories, “Training Algorithm Margin for Optimal Classifiers,” *Perception*, pp. 144–152, 1992.

[158] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Trans. Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[159] S. Wang *et al.*, “A multi-approaches-guided genetic algorithm with application to operon prediction,” *Artif. Intell. Med.*, vol. 41, no. 2, pp. 151–159, 2007.

[160] D. Anand, B. Pandey, and D. K. Pandey, “Knowledge and intelligent computing

techniques in bioinformatics," *Int. J. Comput. Biol. Drug Des.*, vol. 9, no. 3, 2016.

[161] R. Tawil, "Facioscapulohumeral muscular dystrophy.," *Neurotherapeutics*, vol. 5, no. 4, pp. 601–6, 2008.

[162] R. J. L. F. Lemmers, S. O'Shea, G. W. Padberg, P. W. Lunt, and S. M. van der Maarel, "Best practice guidelines on genetic diagnostics of Facioscapulohumeral muscular dystrophy: Workshop 9th June 2010, LUMC, Leiden, The Netherlands," *Neuromuscul. Disord.*, vol. 22, no. 5, pp. 463–470, 2012.

[163] M. B. Eisen, "Cluster analysis and display of genome-wide expression patterns," *Proc. Natl. Acad. Sci.*, vol. 95, no. 25, pp. 14863–14868, 1998.

[164] M. M. Babu, "An introduction to microarray data analysis," *Comput. Genomics Theory Appl.*, pp. 225–249, 2004.

[165] F. Chu and L. Wang, "Applications of support vector machines to cancer classification with microarray data.," *Int J Neural Syst*, vol. 15, no. 6, pp. 475–484, 2005.

[166] D. Anand, B. Pandey, and D. K. Pandey, "Facioscapulohumeral muscular dystrophy diagnosis using hierarchical clustering algorithm and k-nearest neighbor based methodology," *Int. J. E-Health Med. Commun.*, vol. 8, no. 2, 2017.

[167] H. Qian, "A mathematical analysis for the Brownian dynamics of a DNA tether.," *J. Math. Biol.*, vol. 41, no. 4, pp. 331–340, 2000.

[168] D. Anand, B. Pandey, and D. K. Pandey, "An Integrated Algorithm for Dimension Reduction and Classification Applied to Microarray Data of Neuromuscular Dystrophies," *Indian J. Sci. Technol.*, vol. 9, no. 28, 2016.

[169] M. Mitchell, "Genetic algorithms: An overview," *Complexity*, vol. 1, no. 1, pp. 31–39, 1995.

[170] J. Baker, "An algorithm for geometry optimization without analytical gradients," *J. Comput. Chem.*, vol. 8, no. 5, pp. 563–574, 1987.

[171] A. Subasi, "Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders," *Comput. Biol. Med.*, vol. 43, no. 5, pp. 576–586, 2013.

[172] J. Tang, S. Alelyani, and H. Liu, "Feature Selection for Classification: A Review," *Data Classif. Algorithms Appl.*, pp. 37–64, 2014.

[173] Z. M. Hira and D. F. Gillies, "A review of feature selection and feature extraction methods applied on microarray data," *Adv. Bioinformatics*, vol. 2015, no. 1, 2015.

[174] Y. Saeys, I. Inza, and P. Larra??aga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[175] D. Anand, B. Pandey, and D. K. Pandey, "Building an intelligent integrated method of

gene selection for facioscapulohumeral muscular dystrophy diagnosis”, *Int. J. Biomed. Eng. Technol.*, vol. 24, no. 3, 2017.

[176] F. J. Aherne, N. A. Thacker, and P. I. Rockett, “The Bhattacharyya metric as an absolute similarity measure for frequency coded data,” *Kybernetika*, vol. 34, no. 4, pp. 363–368, 1998.

[177] K. G. Babu and M. A. R. Prasad, “An Effective Approach in Face Recognition using Image Processing Concepts”, *International Journal of Application or Innovation in Engineering and Management (IJAIEM)*, vol. 2, no. 8, pp. 215–219, 2013.

[178] A. Sharma, "Review Paper of Various Selection Methods in Genetic Algorithm Types of Selection Method", *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 7, pp. 1476–1479, 2013.

[179] B. Pandey and R. B. Mishra, “An intelligent model for two level diagnoses of neuromuscular diseases,” *International Journal of Knowledge Engineering and Soft Data Paradigm*, vol. 4, no. 3, 2014.

[180] D. Anand, B. Pandey, and D. K. Pandey, “A novel hybrid feature selection model for classification of neuromuscular dystrophies using bhattacharyya coefficient, genetic algorithm and radial basis function based support vector machine,” *Interdiscip. Sci. Comput. Life Sci.*, vol. 10, no. 2, pp. 244-250, 2018.

[181] Z. Cai, R. Goebel, M. R. Salavatipour and G. Lin, "Selecting Dissimilar Genes for multi-class classification, an application in cancer subtyping", *BMC Bioinformatics*, vol. 8, pp. 206, 2007.

[182] R. Rifkin and A. Klautau, “In defense of one-vs-all classification,” *J. Mach. Learn. Res.*, vol. 5, pp. 101–141, 2004.