

Cihan Kuzudışlı

Ph.D. Thesis

AGU 2024

IMPROVING RECURSIVE CLUSTER ELIMINATION APPROACHES FOR GENE EXPRESSION DATA ANALYSIS

Ph.D. THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By

Cihan KUZUDİŞLİ

December 2024

IMPROVING RECURSIVE CLUSTER ELIMINATION APPROACHES FOR GENE EXPRESSION DATA ANALYSIS

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER
ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE OF
ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Cihan KUZUDİŞLİ
December 2024

SCIENTIFIC ETHICS COMPLIANCE

I hereby declare that all information in this document has been obtained in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name-Surname: Cihan KUZUDİŞLİ

Signature :

X X X X

REGULATORY COMPLIANCE

Ph.D. thesis titled “Improving Recursive Cluster Elimination Approaches for Gene Expression Data Analysis” has been prepared in accordance with the Thesis Writing Guidelines of the Abdullah Gül University, Graduate School of Engineering & Science.

Prepared By
Cihan KUZUDİŞLİ
Signature

Advisor
Assoc. Prof. Burcu BAKIR GÜNGÖR
Signature

Co-Advisor
Prof. Dr. Malik YOUSEF
Signature

Head of the Electrical and Computer Engineering Program
Assoc. Prof. Samet GÜLER
Signature

ACCEPTANCE AND APPROVAL

Ph.D. thesis titled “Improving Recursive Cluster Elimination Approaches for Gene Expression Data Analysis” and prepared by Cihan KUZUDİŞLİ has been accepted by the jury in the Electrical and Computer Engineering Graduate Program at Abdullah Gül University, Graduate School of Engineering & Science.

23 /12 / 2024

(Thesis Defense Exam Date)

JURY:

Advisor : Assoc. Prof. Burcu BAKIR GÜNGÖR

Member : Assoc. Prof. Zafer AYDIN

Member : Assoc. Prof. Mete ÇELİK

Member : Assist. Prof. Mehmet Gökhan BAKAL

Member : Assist. Prof. Tayyip ÖZCAN

APPROVAL:

The acceptance of this Ph.D. thesis has been approved by the decision of the Abdullah Gül University, Graduate School of Engineering & Science, Executive Board dated /..... / and numbered

..... / /

(Date)

Graduate School Dean
Prof. İrfan ALAN

ABSTRACT

IMPROVING RECURSIVE CLUSTER ELIMINATION APPROACHES FOR GENE EXPRESSION ANALYSIS

Cihan KUZUDİŞLİ
Ph.D. in Electrical and Computer Engineering Department
Advisor: Assoc. Prof. Burcu BAKIR GÜNGÖR
Co-advisor: Prof. Malik YOUSEF
December 2024

The computational and interpretational difficulties caused by the ever-increasing dimensionality of biological data generated by new technologies pose a significant challenge. Feature selection (FS) methods aim to reduce the dimension, and feature grouping has emerged as a foundation for FS techniques that seek to detect strong correlations among features and identify irrelevant features. In this thesis, methods that utilize feature grouping in a supervised context were developed. We initially tested the effects of different clustering algorithms on SVM-RCE and observed the best performance with K-means. In the first developed method, Recursive Cluster Elimination with Intra-cluster Feature Elimination (RCE-IFE), both cluster and intra-cluster elimination is performed recursively in each cluster reduction step. Our experimental findings imply that RCE-IFE provides robust classifier performance and significantly reduces feature size while maintaining feature relevance and consistency. In the second developed Grouping – Scoring – Model (G-S-M) based study, G-S-M_Rep, we use prior knowledge to form disease groups, and select top features as representative of each group. These representative features are learnt by the model in a cumulative manner. Results show that G-S-M_Rep attains satisfactory model performance with a small number of features. Consequently, this thesis presents methods based on feature grouping and focuses on improving feature reduction capability, classification performance, feature relevancy, and feature consistency.

Keywords: Feature Grouping, Machine Learning, Recursive Cluster Elimination, Intra-cluster Feature Elimination

ÖZET

GEN İFADE MİKTARI VERİSİ ANALİZİ İÇİN YİNELEMELİ ÖBEK ELİMİNASYON YÖNTEMLERİNİN İYİLEŞTİRİLMESİ

Cihan KUZUDİŞLİ

Elektrik ve Bilgisayar Mühendisliği Anabilim Dalı Doktora

Tez Danışmanı: Doç. Dr. Burcu BAKIR GÜNGÖR

İkinci Tez Danışmanı: Prof. Dr. Malik YOUSEF

Aralık 2024

Yeni teknolojilerle üretilen biyolojik verilerin giderek artan boyutluluğunun neden olduğu hesaplama ve yorumlama güçlükleri önemli bir zorluk oluşturmaktadır. Özellik seçimi (FS) yöntemleri boyutu azaltmayı amaçlar ve özellik gruplaması, özellikler arasında güçlü korelasyonları tespit etmeyi ve ilgisiz özellikleri belirlemeyi amaçlayan FS teknikleri için bir temel olarak ortaya çıkmıştır. Bu tezde, gözetimli bir bağlamda özellik gruplandırmasını kullanan yöntemler geliştirilmiştir. Başlangıçta farklı kümeleme algoritmalarının SVM-RCE üzerindeki etkilerini test ettik ve K-means ile en iyi performansı gözlemledik. Geliştirilen ilk yöntem olan Öbek İçi Özellik Eleme ile Yinelemeli Öbek Eleme (RCE-IFE) yönteminde, her öbek azaltma adımında hem öbek hem de öbek içi eleme yinelemeli olarak gerçekleştirilir. Deneysel bulgularımız, RCE-IFE'nin güçlü bir sınıflandırıcı performansı sağladığını ve özellik ilgisini ve tutarlılığını korurken özellik boyutunu önemli ölçüde azalttığını göstermektedir. İkinci geliştirilen Gruplama – Puanlama – Model (G-S-M) tabanlı çalışma olan G-S-M_Rep'de, hastalık gruplarını oluşturmak için ön bilgileri kullanıyoruz ve her grubu temsil edecek en iyi özellikleri seçiyoruz. Bu temsili özellikler model tarafından kümülatif bir şekilde öğrenilir. Sonuçlar G-S-M_Rep'in az sayıda özellik ile tatmin edici bir model performansına ulaştığını göstermektedir. Sonuç olarak, bu tez özellik gruplandırmaya dayalı yöntemleri sunmakta ve özellik azaltma yeteneğini, sınıflandırma performansını, özellik alaka düzeyini ve özellik tutarlılığını iyileştirmeye odaklanmaktadır.

Anahtar kelimeler: Özellik Gruplandırma, Makine Öğrenimi, Yinelemeli Öbek Eliminasyonu, Öbek İçi Özellik Eleme

Acknowledgements

I would like to express my deepest gratitude to my advisor, Assoc. Prof. Burcu Bakır Güngör, for her invaluable guidance, unwavering support, and insightful critiques throughout this journey. Her mentorship played a pivotal role in the successful completion of this thesis study.

I extend my sincere thanks to my co-advisor, Prof. Malik Yousef, for his constructive feedback and essential suggestions that greatly enriched the quality of my work. His wisdom and encouragement impacted and inspired me throughout my academic journey.

I also thank Prof. Bahjat Qaqish for his continuous support and valuable advice. His availability in our weekly meetings contributed significantly to my academic development.

I am also grateful to my family and friends for their assistance in handling the challenges during this process.

My special thanks to my wife, Fatma Büşra Kuzudişli, for her patience, smiling face, and constant support through this research. I would also like to thank my son, Abdulkadir Kuzudişli, for all the recreation and emotional support.

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1 NOVELTY AND CONTRIBUTIONS OF RESEARCH	2
1.2 BASICS OF FEATURE SELECTION	4
1.1.1 Filter methods	4
1.1.1.1 Information gain.....	5
1.1.1.2 Pearson's correlation.....	6
1.1.1.3 Chi square	6
1.1.1.4 Mutual information	7
1.1.1.5 Symmetrical uncertainty.....	7
1.1.2 Wrapper methods	8
1.1.3 Embedded methods	9
1.2 FEATURE SELECTION APPROACHES.....	10
1.2.1 Traditional feature selection.....	10
1.2.2 Feature selection through feature grouping	11
1.2.2.1 Grouping-based feature selection under supervised setting	11
1.2.2.2 Grouping-based feature selection under unsupervised setting	16
1.2.2.3 Grouping-based feature selection under semi-supervised setting	18
1.2.3 Grouping features with biological domain knowledge	19
1.2.4 Feature grouping with recursive cluster elimination	23
2. EFFECT OF DIFFERENT CLUSTERING ALGORITHMS ON SVM-RCE. 27	
2.1 CLUSTERING ALGORITHMS.....	27
2.1.1 K-means algorithm.....	27
2.1.2 K-medoids clustering	28
2.1.3 Hierarchical clustering	29
2.1.4 Gaussian Mixture Model clustering.....	29
2.2 EXPERIMENTAL SETUP	30
2.3 RESULTS.....	31
2.4 DISCUSSION.....	33
3. RCE-IFE: RECURSIVE CLUSTER ELIMINATION WITH INTRA-CLUSTER FEATURE ELIMINATION	34
3.1 MATERIALS AND METHODS.....	34
3.1.1 Datasets.....	34
3.1.2 Proposed Algorithm	35
3.2 RESULTS.....	43
3.2.1 Experimental Setup	43
3.2.2 Comparative performance evaluation of RCE-IFE with SVM-RCE on gene expression, miRNA, and methylation datasets	44
3.2.3 Comparative performance evaluation of RCE-IFE with SVM-RCE on metagenomics datasets	46
3.2.4 Comparative performance evaluation of RCE-IFE with conventional FS methods.....	47
3.2.5 Comparative performance evaluation of RCE-IFE with Multi-stage algorithm	50

3.2.6 Comparative performance evaluation of RCE-IFE with other biological domain knowledge-based G-S-M tools.....	50
3.2.7 Biological validation of the RCE-IFE findings.....	51
3.2.8 Consistency of the selected features across different runs	53
3.3 DISCUSSION.....	53
4. DETERMINING REPRESENTATIVE FEATURES VIA GROUPING, SCORING AND MODELING (G-S-M)	55
4.1 DATASETS	55
4.2 THE PROPOSED G-S-M BASED APPROACH	56
4.3 RESULTS.....	56
4.3.1 Comparison of G-S-M_Rep on gene expression datasets.....	58
4.3.2 Pathway analysis.....	59
4.3.3 Hub gene analysis	59
4.3.4 Gene analysis for GDS2545 dataset	60
4.4 DISCUSSION.....	60
5. CONCLUSIONS AND FUTURE PROSPECTS	62
5.1 CONCLUSIONS	62
5.2 SOCIETAL IMPACT AND CONTRIBUTION TO GLOBAL SUSTAINABILITY	64
5.3 FUTURE PROSPECTS	65

LIST OF FIGURES

Figure 1.1 Three basic types of FS methods. (A) Filter (B) Wrapper (C) Embedded.....	5
Figure 1.2 The representation of feature selection approaches based on grouping.....	12
Figure 1.3 Typical approach for representative feature selection based on grouping	12
Figure 1.4 The workflow of the SVM-RCE algorithm. The grouping step (G) for grouping genes into clusters, the scoring step (S) for assigning score for each cluster and selecting significant clusters, the modeling step (M) for training the model with top-ranked clusters.	24
Figure 2.1 The effects of clustering algorithms on execution time per iteration.....	32
Figure 3.1 The workflow of RCE-IFE algorithm	40
Figure 3.2 Extraction of subdatasets based on gene clusters	41
Figure 3.3 Assigning a score to a subdataset	42
Figure 3.4 Running time comparison of RCE-IFE, RCE-IFE-SVM and SVM-RCE on miRNA and methylation datasets	46
Figure 3.5 Comparative performance evaluation of RCE-IFE with three biological domain knowledge-based G-S-M tools	52
Figure 3.6 Overlaps between the top (A) 5, (B) 10, (C) 20 miRNAs selected in different runs.....	53
Figure 4.1 Creating feature groups using prior knowledge	56
Figure 4.2 The framework of the G-S-M_Rep approach.....	57
Figure 4. 3 Pathway – gene network of all datasets.....	59
Figure 4. 4 Heatmap of genes through 10 iterations.....	61

LIST OF TABLES

Table 2.1 Basic information on gene expression datasets	31
Table 2.2 Average gene and AUC values for four clustering algorithms.....	31
Table 3.1 Basic information about the gene expression, miRNA, and methylation datasets	36
Table 3.2 Basic information about the metagenomics datasets	36
Table 3.3 Basic information about the cancer-related datasets	37
Table 3.4 Comparative performance evaluation of RCE-IFE with SVM-RCE: Mean AUC and mean size of the feature subsets on gene expression, miRNA and methylation datasets.....	45
Table 3.5 Comparative performance evaluation of RCE-IFE with SVM-RCE: Mean AUC and mean size of the feature subsets on metagenomics datasets	47
Table 3.6 Mean AUC values after comparing with popular FS algorithms for the same number of selected genes.....	48
Table 3.7 Comparative performance evaluation of RCE-IFE with SVM-RCE: Mean AUC and mean size of the feature subsets on metagenomics datasets	50
Table 4.1 Main characteristics of 10 gene expression datasets	55
Table 4.2 Comparison between standard G-S-M and G-S-M_Rep for top two groups .	58
Table 4.3 Gene counts for standard G-S-M and G-S-M_Rep methods for top five groups	58
Table 4. 4 Number of hub genes in the network of Standard G-S-M genes.....	60

LIST OF ABBREVIATIONS

ACA	Attribute Clustering Algorithm
AHC	Agglomerative Hierarchical Clustering
ASD	Autism Spectrum Disorder
AUC	Area Under Curve
BAS	Binary Ant System
BRCA	Breast Cancer
CART	Classification and Regression Tree
CLARANS	Clustering Large Applications based on RAN-domized Search
CMIM	Conditional Mutual Information Maximization
CRC	Colorectal Cancer
DCG-Net	Distance Correlation Gain-Network
DHC	Divisive Hierarchical Clustering
DT	Decision Tree
ENA	European Nucleotide Archive
FAST	Fast clustering-bAsed feature Selection algoriThm
FCBF	Fast Correlation-Based Filter
FCLARANS	Fuzzy Clustering Large Applications based on RAN-domized Search
FCM	Fuzzy C-Means
FS	Feature Selection
G	Grouping
GDC	Genomic Data Commons
GMM	Gaussian Mixture Model
GO	Gene Ontology
GreedyPGG	Greedy Predominant Groups Generator
G-S-M	Grouping-Scoring-Modeling
GXNA	Gene eXpression Network Analysis
HC	Hierarchical Clustering
IBD	Inflammatory Bowel Disease
IBDMDB	Inflammatory Bowel Disease Multi-omics Database
IFE	Internal Feature Elimination

IG	Information Gain
KEGG	Kyoto Encyclopedia of Genes and Genomes
KNN	K-nearest Neighbors
LDA	Latent Dirichlet Allocation
M	Modeling
MCCV	Monte Carlo Cross-Validation
MI	Mutual Information
MICAP	Maximal Information Coefficient and Affinity Propagation
miRNA	microRNA
ML	Machine Learning
MLE	Maximum Likelihood Estimation
MRMR	Minimum Redundancy Maximum Relevance
MST	Minimum Spanning Tree
NA- β -PFRS	Neighborhood Adaptive β -precision Fuzzy Rough Set
NCI	National Cancer Institute
PCS	Postconcussion Syndrome
PSO	Particle Swarm Optimization
PTSD	Posttraumatic Stress Disorder
RCE-IFE	Recursive Cluster Elimination with Intra-cluster Feature Elimination
RF	Random Forest
S	Scoring
SAM	Significance Analysis of Microarrays
SDG	Sustainable Development Goal
SKB	SelectKBest
SNR	Signal to Noise Ratio
SRA	Sequence Read Archive
SU	Symmetrical Uncertainty
SVC	Support Vector Classifier
SVM-RCE	Support Vector Machine - Recursive Cluster Elimination
SVM-RFE	Support Vector Machine - Recursive Feature Elimination
SVM-RNE	Support Vector Machine - Recursive Network Elimination
T2D	Type 2 Diabetes

VNS

Variable Neighborhood Search

XGBoost

eXtreme Gradient Boosting





To my family

Chapter 1

Introduction

In the current digital era, the data produced by many applications in fields such as image processing, pattern recognition, machine learning (ML) and network communication grow exponentially in both dimension and size. Due to this high-dimensionality, the search space is widening and extraction of valuable knowledge from the data becomes a challenging task [1-2]. Also, utilizing all features in a dataset is unlikely to develop a predictive model with high accuracy. The existence of irrelevant and redundant features may weaken the generalizability of the model and decrease the overall precision of a classifier [3-4]. Hence, reducing the number of input variables is highly desired as it lowers the computational cost of model construction and allows improving model performance. As such, feature selection (FS) becomes an inevitable step for domain experts and data analysts.

FS is the process of selecting the minimally sized feature subset from the original set that is optimal for the target concept. It plays a crucial role in removing irrelevant and redundant features while keeping relevant and non-redundant ones [5]. Irrelevant features do not alter the target concept in any way and redundant features do not contribute to the target concept [6]. These features may contain a considerable amount of noise which can be misleading, resulting in significant computational overhead and poor predictor performance. Contrary to other dimensionality reduction techniques, FS preserves the data semantics as it does not distort the original feature representation and hence provides straightforward data interpretation for data scientists. Additionally, reduction in dimension by FS prevents overfitting that can lead to undesired validation results.

Although various FS techniques have been developed, traditional approaches to FS neglect structures of features during the selection process. Another issue is that retention and elimination of features on an individual basis ignores dependence among them. Because of these reasons, correlation between features may not be detected efficiently resulting in irrelevant or redundant features in the final subset. Some studies grouped samples (i.e., observations) for improving classification performance but these studies

were not concerned with feature reduction at all [7-8].

On the other hand, FS based on grouping is an effective technique for reducing feature redundancy and enhancing classifier learning. By grouping the features, the search space is reduced substantially. Moreover, it can reduce estimator variance [9], improve stability, and reinforce generalization capability of the model. In this procedure, the process of grouping features into clusters is generally performed as the initial step, aiming to have maximal intra-class similarity (i.e., similarity in between the objects of the same cluster) and minimal inter-class similarity (i.e., objects in a cluster are more similar to those in another one) between features. These feature groups can be created by K-means, fuzzy c-means (FCM), hierarchical clustering, graph theory and other methods [10-13]. After cluster formation, features within each cluster are scored and selected using various techniques or metrics.

1.1 Novelty and Contributions of Research

The research aim is to develop approaches that provide robust outcomes in terms of identified genes and clusters, which will in turn pave the way for prediction, diagnosis, and treatment of diseases as well as understanding mechanisms of their onset and progression. In addition, the research is aimed at improving the classifier performance or avoid poor performance while reducing the number of features. This thesis presents the designed ML algorithms based on grouping of features (i.e., feature clusters) and evaluates the performance and feature reduction capability.

This chapter gives the literature review on FS approaches based on feature grouping. In FS-based feature grouping, the aim is to first keep similar features together within clusters while maximizing diversity between clusters followed by selection of features out of clusters. We can conclude that sequential and optimization-based (i.e., fuzzy and evolutionary) FS approaches are noteworthy in this context since they take feature interactivity into consideration during the selection phase. Hybrid approaches or utilizing a combination of different techniques are also effective because each method brings its advantage. In the case of biological data, integrating external knowledge can yield better results in the overall analysis. In fact, the availability of independent and relevant features, correlation between features, and feature correlation to the decision are important items to be taken into consideration. The models with the ability to take these factors into consideration are likely to be effective in FS.

In Chapter 2, the effect of different clustering algorithms on Support Vector Machine – Recursive Cluster Elimination (SVM-RCE) algorithm is presented. SVM-RCE involves a clustering step which originally is k-means. Using various performance metrics, three alternative algorithms are evaluated in this context; k-medoids, Hierarchical Clustering (HC), and Gaussian Mixture Model (GMM). Comparisons are carried out on five publicly available gene expression datasets. The results show that k-means in SVM-RCE obtains higher performance than other tested algorithms in terms of classification performance. Additionally, HC shows a similar performance to k-means. The findings show superiority of using k-means.

The developed approach, Recursive Cluster Elimination with Intra-cluster Feature Elimination (RCE-IFE), is covered in Chapter 3. RCE-IFE, an extension of SVM-RCE, integrates feature removal within surviving clusters at each step of cluster reduction. Through this newly added phase, less contributing features in each cluster are excluded, leading to deeper dimension reduction and also improvements in feature subset quality, classification performance, and running time. We evaluate the dimension reduction capability and predictive power of RCE-IFE on various datasets. For a combination of gene expression, microRNA (miRNA) expression, and methylation datasets, RCE-IFE achieves an average area under the curve (AUC) of 0.85 using the least number of features with the least execution time, while RCE-IFE-SVM (SVM-adapted version of RCE-IFE) and SVMRCE generate similar average AUCs of 0.84 and 0.83, respectively. When tested on seven different metagenomics datasets, RCE-IFE and SVM-RCE yields an average AUC of 0.79 and 0.68 respectively, while RCE-IFE providing a remarkable reduction in feature subsets.

Moreover, RCE-IFE predominates among several state-of-the-art FS methods, including MRMR, FCBF, IG, CMIM, SKB, and XGBoost, by providing an average AUC of 0.76 on five gene expression datasets. In comparison with Multi-stage algorithm, RCE-IFE yields a similar accuracy rate of 89.27%, averaged over four cancer-related datasets using a smaller number of features. Additionally, we have comparatively evaluated RCE-IFE with other biological domain knowledge-based Grouping-Scoring-Modeling (G-S-M) tools (mirGediNET, 3Mint, and miRcorrNet) and conducted biological validation of the selected features. We further show the high consistency of the features selected by RCE-IFE across multiple runs. Overall, via testing through diverse biological datasets concerning various diseases, the experimental findings show the effectiveness of the proposed RCEIFE method with respect to predictive power, reduced feature subset

selection, feature relevancy, and consistency of selected features across different runs.

Next, the designed Grouping-Scoring-Modeling (G-S-M) based approach is explained in Chapter 4. In this methodology, named as G-S-M_Rep, we incorporate knowledge domain into grouping of features rather than statistical approaches. We compared G-S-M_Rep with Standard G-S-M and obtained similar classification performance with substantially reduced feature number. We also performed pathway analysis and presented the related pathways although datasets belong to different type of diseases. We also conducted the analysis of determination of hub genes. Finally, we examined the gene patterns across iterations to deduce significant genes. The research is evaluated and summarized in Chapter 5 with future prospects.

1.2 Basics of Feature Selection

In this section, we present basic concepts in FS. According to their interaction with the classification model, FS techniques can be classified into filter, wrapper, and embedded techniques [14]. Later in the literature, hybrid and ensemble techniques have emerged as variants of them. Hybrid approach combines two different methods to utilize the advantages of both approaches, where the common combination is filter and wrapper methods. Ensemble technique integrates an ensemble of feature subsets and then yields the result from the ensemble. The overview of the three main types of methods is shown in Figure 1.1.

1.1.1 Filter methods

Filter type methods select features by assessing intrinsic properties of data based on statistical measures instead of cross-validation performance. They are easily scalable to high-dimensional datasets, independent of the learning algorithm; they are simple and computationally fast; and they are resistant to overfitting. In this method, each feature is assigned a score determined by the selected statistical method. Afterwards, all features are ranked in descending order and those with low scores are removed using a threshold value. The remaining features comprise the feature subset and are then fed into the classification model. Consequently, FS is carried out once and then various classifiers can be employed. Disadvantages of this technique are (i) features are selected irrespective of

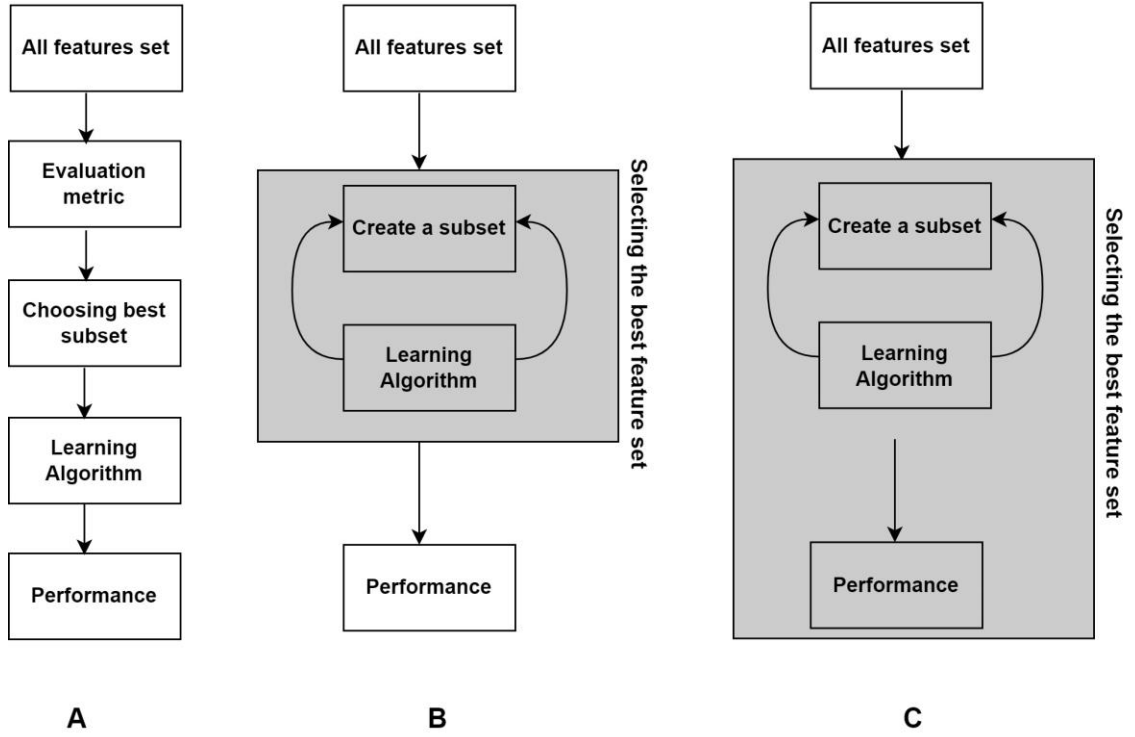


Figure 1.1 Three basic types of FS methods. (A) Filter (B) Wrapper (C) Embedded

the classifier, and (ii) feature dependencies are ignored. Some common statistical measures used in this technique are information gain (IG), Pearson's correlation, Chi square (χ^2), mutual information (MI), and symmetrical uncertainty (SU).

1.1.1.1 Information gain

Information gain (IG) [15] is an entropy-based FS method and used to measure how much information a feature carries about the target variable. IG of a feature X in a data group D with n class labels, $IG(X)$, is calculated using

$$IG(X) = E(D) - \sum_{i=1}^n \frac{D_i}{D} E(D_i) \quad (1.1)$$

where $E(D)$ denotes the general entropy belonging to class labels, D_i/D is the ratio of number of occurrences of each value on feature X , and $E(D_i)$ specifies the entropy of i th feature value calculated by splitting dataset D based on feature X . Entropy is a measurement of unpredictability or impurity of a data distribution and defined as:

$$E(D) = - \sum_{i=1}^n p(i) \log_2 p(i) \quad (1.2)$$

where $p(i)$ is the probability of class i in the data group D for n class labels. A feature is relevant to target variable if it has a high information gain. The way the features are selected is in a univariate way (i.e., features are selected independently), therefore, redundant features cannot be eliminated in this technique.

1.1.1.2 Pearson's correlation

Pearson's correlation is a measure of the dependency (or similarity) of two variables and used for finding the relationship between continuous features and the target feature [16-17]. It produces the correlation coefficient r ranging between -1 to 1, where 1 shows a strong correlation and -1 means a total negative correlation. So, 0 value implies no correlation between the features. A positive correlation states that if one variable increases, so does the other variable, whereas a negative correlation implies that while one variable raises, another one decreases. This method can also be used to measure correlation between pairs of features. In this way redundant features can be identified. Pearson's correlation coefficient r can be found for feature X with values x and classes Y with values y where X, Y are random variables by the following equation:

$$r = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2 \sum(y - \bar{y})^2}} \quad (1.3)$$

where $\bar{x}$ and $\bar{y}$ are means of x and y , respectively. Note that Pearson's correlation is mainly covariance of two variables divided by product of their standard deviations.

1.1.1.3 Chi square

Chi square (χ^2) [18] is a statistical method to test the independence of two events. It is a measurement of the degree of association between two categorical values. It measures the deviation from the expected frequency assuming the feature event is independent of the class label. This assumption is tested for a given feature with n class and m different feature values by the formula

$$\chi^2 = \sum_{i=1}^m \sum_{j=1}^n \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \quad (1.4)$$

where O_{ij} is the observed (i.e., actual) value and E_{ij} refers to the expected value suggested by the null hypothesis. E_{ij} is calculated as

$$E_{ij} = \frac{(O_{*j}O_{i*})}{O} \quad (1.5)$$

where O_{*j} means the number of samples in class m , and O_{i*} indicates the number of samples with the i^{th} feature value for the feature under study. Higher value of χ^2 shows rejection to the null hypothesis, namely, higher dependency between the feature and the class label.

1.1.1.4 Mutual information

Mutual information (MI) [19] is another statistical method used to assess the mutual dependence between the two variables. MI quantifies the amount of information that one random variable includes in the other random variable. MI between two continuous random variables X and Y with their joint probability functions $p(x, y)$, and their marginal probability density functions $p(x)$ and $p(y)$, respectively is given by

$$I(X; Y) = \iint p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dx dy \quad (1.6)$$

For discrete random variables, the double integral is substituted by a summation as

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \quad (1.7)$$

We can also define the conditional mutual information (CMI) of two random variables X and Y given a third variable Z as

$$I(X; Y|Z) = \iiint p(x, y, z) \log \frac{p(x, y|z)}{p(x|z)p(y|z)} dx dy dz \quad (1.8)$$

It can be interpreted as the amount of information X includes in Y which is not shared by Z .

1.1.1.5 Symmetrical uncertainty

This is one of the techniques that are used to measure redundancy between two random variables [20]. It is obtained by normalizing MI to the entropies of two variables and limiting it to the range of [0,1]. It's able to circumvent inherent bias of MI toward features with a wide range of different values. Symmetrical uncertainty (SU) is defined as

$$SU(X, Y) = \frac{2MI(X, Y)}{H(X) + H(Y)} \quad (1.9)$$

where $H(X)$ and $H(Y)$ are entropy of variable X and Y , respectively. A value 1 between a pair of features indicates that knowledge of feature value can fully predict the values of other and 0 value shows that X and Y are not correlated.

Based on SU , C-Relevance between a feature and a target variable C , and F-Correlation between feature pair can be defined as follows [21]:

C-Relevance: SU between feature $F_i \in F$ and target variable C , denoted by $SU_{i,c}$.

F-Correlation: SU between any feature pair F_i and F_j ($i \neq j$), denoted by $SU_{i,j}$.

1.1.2 Wrapper methods

In this methodology, a search strategy for possible subsets of features is defined, and the learning algorithm is trained using these subsets in an iterative manner. Unlike filter methods, wrapper methods are in interaction with the classifier, however, the evaluation of feature subsets is obtained using a specific classification model which makes this method specific to a learning model. Several possible combinations of features are evaluated in the model by wrapping the search algorithm around it [22]. This method provides suboptimal feature subsets for training the model since evaluating all possible subsets is computationally not practical, and generally gives better predictive accuracy than filter methods but is computationally intensive due to searching overhead and learner dependence.

The search for generating subsets can be performed with schemes such as forward selection, backward elimination, stepwise selection or a heuristic search [23]. Forward selection is a repetitive technique where no feature is considered at the onset. Initially, the feature with the best performance is added. Then another more significant feature giving the best performance together with the previously added feature is selected. This process proceeds until the inclusion of a new feature does not improve the classifier performance. In backward elimination, the algorithm starts with all the features available and discards the most insignificant feature from the model recursively. This elimination process is repeated until removal of features does not enhance the performance of the model. For stepwise selection, this technique is a combination of both forward selection and backward elimination. It starts with an empty set and the most significant feature is added at each iteration. While adding a new feature, previously selected features are removed if any of them has become insignificant. Heuristic search is concerned with

optimization and aims at optimizing the objective function in evaluation of different subsets [24].

Support Vector Machine - Recursive Feature Elimination (SVM-RFE) [25] is a popular example of wrapper methods. The idea is mainly to train the classifier by the given data and assign a rank by SVM for each feature as its weight. Then, features with the smallest weights are removed by a specific rate determined by the user. This procedure is repeated until reaching a predefined number of features.

1.1.3 Embedded methods

This method includes advantages of filter and wrapper methods and performs FS and model construction at the same time. Just like wrapper techniques, they are specific to a learning model but they have less computational complexity than wrapper methods [26]. One technique of this type of FS is regularization that adds a penalty to the coefficients to overcome overfitting in the model. As an example, Lasso [27] is an embedded method that uses L_1 norm of the coefficient of a linear classifier $\mathbf{w}$ and penalty term (Φ) is defined as

$$\Phi(\mathbf{w}) = \sum_{i=1}^k |\mathbf{w}_i| \quad (1.10)$$

and

$$\hat{\mathbf{w}} = \min_{\mathbf{w}} c(\mathbf{w}, X) + \alpha \Phi \quad (1.11)$$

where $c(\cdot)$ is the objective function for classification, Φ is a regularization term, k is the number of features, α is the regularization parameter controlling the trade-off between the objective function and the penalty. These coefficients may even be reduced to 0 for features that do not contribute to the model. Features with non-zero coefficients are retained and those with low or zero coefficient are excluded [28]. Another technique to integrate FS in model creation is decision trees. These tree-based methods are non-parametric models that consider features as nodes. Tree-based strategies used by random forests accumulate various numbers of decision trees and rank the nodes (i.e., features) by decrease in the impurity (e.g., Gini impurity) over all the trees, e.g., classification and regression tree (CART) [29].

1.2 Feature Selection Approaches

Broadly speaking, FS algorithms conducted in many studies can be categorized into the following two classes: (i) traditional FS, (ii) FS based on grouping. Traditional approaches generally consider all features contingent on “singularity” during the selection process. To put it another way, they comprise inclusion or elimination of features based on some statistical measures or classifying capacity at a singular level. On the other hand, grouping-based methods detect relevant features by grouping them into clusters; and then remove redundant ones which lead to reduced search space.

1.2.1 Traditional feature selection

Different FS methods exist in abundance in the literature, including filters based on distinct criteria such as dependency, information, distance and consistency [30], and wrapper and embedded methods employing different induction algorithms. Due to their simplicity, filter methods are often preferable in the context of high dimensional data; the absence of necessity for a search route and the interaction with a classifier makes them computationally efficient and practically feasible in applications. A comparative study on various filtering methods including mixture model, regression modeling and t -test was presented in [31] where the authors outlined similar and dissimilar aspects of these methods. The authors noted that all the three methods employ two-sample t -test or its variation; but these methods vary in different significance levels and the number of detected features. Lazar et al. [32] also reviewed filter type FS algorithms used in gene expression data analysis and presented them as a top-bottom strategy in a taxonomy.

Wrapper methods carry the computational burden since they require navigation in the search domain and since they interact with the predictor. However, they provide better accuracy than filter approaches due to their interaction with the learning algorithm. Talavera [33] compared filter and wrapper approaches in clustering. They confirm the superiority of wrappers along with some of their problems and they suggest filter techniques as an alternative approach due to their computational efficiency. A recent study [34] overviewed existing wrapper techniques and evaluated the pros and cons of them. Embedded methods, like wrapper techniques, possess computational complexity when it comes to high-dimensional data. They are more efficient than wrappers and have less complexity. Applications of this approach in the bioinformatics domain have been

reviewed in [35].

Hybrid methods combine two methods such as filter and wrapper to take advantage of both methods in order to increase efficiency and performance. Ensemble methods integrate different methods for FS, classification or both. In this approach, multiple feature selectors, induction algorithms, different subsets may be included according to the design scheme. A detailed discussion on hybrid methods and a good review on ensemble FS techniques can be found in [36] and [37], respectively. In some studies, FS methods are divided into these five categories [38].

Traditional FS approaches have several shortcomings. For instance, filter methods evaluate the significance of each feature individually without considering the relationships and interactions between the features. Wrapper methods can provide the optimal feature subset but their complexity makes them imperfect, they are not preferable especially in combinatorial techniques such as in ensemble methods. In addition, they are not applicable to data with small number of samples due to overfitting. Embedded methods, like wrappers, are specific to the model hence may give a different feature subset for the same dataset. The main drawback behind such methods is their inability to remove redundant features and retain informative features efficiently [39-40].

1.2.2 Feature selection through feature grouping

In this section, we will categorize FS approaches based on feature grouping under supervised, unsupervised and semi-supervised context. Supervised FS utilizes data labels to measure importance and relevance of features. Unsupervised FS, on the other hand, assesses feature relevance by exploiting natural structure of the data without using the class label. Semi-supervised FS benefits from both labeled and unlabeled data. Figure 1.2 illustrates a taxonomy of grouping-based FS approaches covered in this section. A typical scenario in FS approaches based on grouping is that the features are first partitioned into clusters and then (a) representative feature(s) is (are) selected from each cluster according to a specific metric or technique as shown in Figure 1.3.

1.2.2.1 Grouping-based feature selection under supervised setting

In the literature, there are many studies that conducted FS through feature grouping. The grouping of features is performed by various techniques including K-means [41], hierarchical clustering [42-43], affinity propagation [44], graph theories [45], information theory metrics [46], kernel density estimation [47], logistic regression [48], and

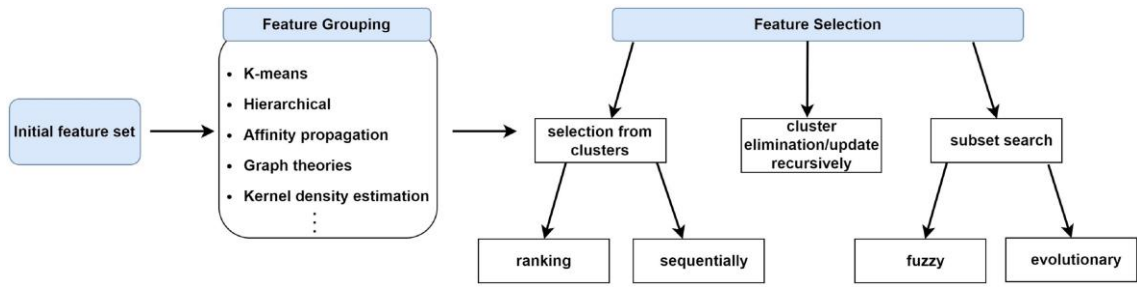


Figure 1.2 The representation of feature selection approaches based on grouping

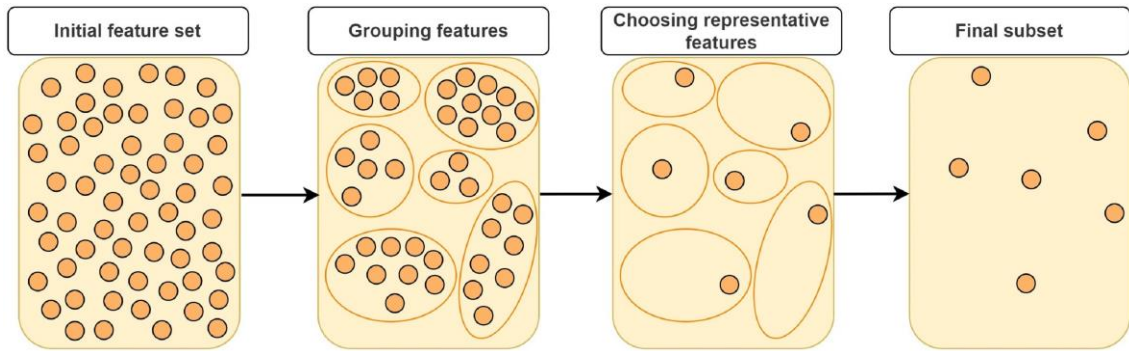


Figure 1.3 Typical approach for representative feature selection based on grouping

regularization methods [49]. With the availability of class labels in datasets, this prevalence is increasing day by day, offering new approaches and gaining new insights into the field.

Several studies performed K-means or hierarchical clustering for grouping features and then they chose genes from each cluster. Sahu et al. [50] proposed an ensemble approach where K-means is applied first for feature grouping and then three different filter-based ranking techniques (t -test, signal-to-noise ratio (SNR) and significance analysis of microarrays (SAM)) are implemented for each cluster independently; and the feature in the front rank from each cluster is selected to form three distinct feature subsets. Afterwards, features in subsets are subject to additional elimination by checking the inclusion of each feature in other subsets. In other words, a feature is discarded if it is not available in other subsets. They obtain good accuracy for different combinations in general but this study ignores correlations between genes. Another study [51] applied information compression index to group features by hierarchical clustering and then sorted features within each cluster by Fisher criterion measuring the classification capacity of each feature in a cluster. Subsequently, the feature in the front rank is selected for each cluster to form the feature subset.

Regarding selection of features from groups, in addition to ranking, selection can also be performed sequentially. For instance, Zhu and Yang [52] group features into clusters by a modified affinity propagation algorithm, and then they apply sequential FS for each cluster. Later on, they gather selected features in clusters to acquire the reduced subset. Their experimental results show improvement in execution time and the accuracies are comparable with sequential FS. Alimoussa et al. [53] proposed a sequential FS method based on feature grouping mainly consisting of three steps. They first remove irrelevant features using Pearson correlation. Then, the same correlation metric is employed for grouping of features into clusters by considering intercorrelated features directly or indirectly via other features. Finally, a feature from each cluster is selected sequentially and features belonging to the same cluster are removed in each round. Their proposed method gives better accuracy and reduction in size compared to filter and wrapper methods. However, despite their approach being fully filter-based, execution time of the proposed method is moderate due to the grouping procedure. In their other work for color texture classification [54], they incorporate a classifier into their previous work in order to measure accuracy when a feature is added at each step of their procedure, thereby determining the dimensionality of the feature subset. They show that combining several descriptor configurations performs better compared to a predefined configuration.

Au et al. [55] proposed an effective algorithm called k-modes attribute clustering algorithm (ACA) for gene expression data analysis. This algorithm uses an information measure to quantify correlation between features, and performs K-mode algorithm, similar to K-means, to cluster features. They defined mode of each cluster as the attribute (i.e., feature) with the highest sum of relevancy with others in each feature group. These modes constituted the final reduced subset. Their measure was also utilized to get good clustering configurations automatically. Chitsaz et al. [56] presented a fuzzy variant of this study which relies on the basic underlying idea in fuzzy clustering approaches, that each feature may belong to more than one group. Rather than considering association of each feature with a sole cluster, association with all features among the overall clusters is considered by assigning different grades of membership to features. Their extended work [57] integrates chi-square test to assess the dependency of each feature on the class labels during the FS process. In their method, objective function is computed by the following formula

$$J = \sum_{r=1}^k \sum_{i=1}^p u_{ri}^m R(A_i; \eta_r) \quad (1.12)$$

where k and p designate number of clusters and features, respectively and u_{ri} is membership degree of i th feature in r th cluster and m is a weighting exponent with η_r being the mode of r th cluster which is essentially center of that cluster. R function denotes interdependence measure between feature A_i and mode η_r . Their experimental results achieve improvement in the accuracy of the classifier with significant reduction in selected feature size compared to the basic version.

Graph-based approaches are also common in studies involving FS through grouping. Song et al. [21] proposed an algorithm, called Fast clustering-based feature Selection algorithm (FAST), and benefited from minimum spanning trees (MST) to create feature clusters. They adopted SU to determine relevance between any pair of features or between the feature and the target class. Finally, the feature with the highest correlation with the class label is selected from each cluster. Another study [58] under supervised framework similarly used MST for grouping and variation of information for relevance measure. Desired number of features and the pruning rate should be given as inputs in their algorithm. A recent study by Zheng et al. [59] builds the graph by interaction gain, makes use of MST to produce feature groups and probabilistic consistency measure for quality metric including two different techniques for FS: in the first one, they apply the conventional way of selecting representatives from each feature groups; and in the second they use harmony search as a metaheuristic search. The metaheuristic approach dominates their first proposed algorithm together with other search mechanisms. Quite recently, the study proposed by Wan et al. [60] employs graph theory for feature grouping and selection in a fuzzy space. They initially construct the fuzzy space using neighborhood adaptive β -precision fuzzy rough set (NA- β -PFRS) and then constitute feature groups using MST and acquire the final subset considering feature to feature and feature-to class relevance in the space. They achieve slightly better results in accuracy with reduced number of features in comparison with other FS approaches and they also show robustness of their model.

Speaking of metaheuristic, García-Torres et al. [61] employed Markov blanket for clustering features and then these predominant groups are involved in variable neighborhood search (VNS) metaheuristic. Their algorithm yields competitive results in classifier performance and exhibits effective results in terms of number of features and

running time. Another optimization-based approach in [62] adopted a scatter search (SS) strategy based on feature grouping where Greedy Predominant Groups Generator (GreedyPGG) [61] is used to group features. In their metaheuristic approach, each solution generated by the search is enhanced with sequential forward selection for selection of the reduced set of features. Their experimental work shows comparable classification results with SS but a significant reduction in feature subset size. Song et al. [63] presents a three-step hybrid study for high dimensional data. Their work initially removes irrelevant features with SU by a predetermined threshold ρ_0 which is defined as

$$\rho_0 = \min (0.1 \times SU_{max}, SU_{\lfloor D/\log D \rfloor - th}) \quad (1.13)$$

where SU_{max} is the maximal relevance value between a feature and class labels among all D features. Secondly, it constitutes feature groups using a SU-based clustering approach in which cluster centers are chosen at first and initial number of clusters is not required. As the third step, representative features are selected from clusters based on particle swarm optimization (PSO) with global search capability. Their proposed methodology yields comparative results with respect to accuracy and running time. García-Torres et al. extended their previous SS work in [64], integrating an additional stopping criterion into their algorithm along with hyperparameter tuning. Their experimental results present the effectiveness of the additional stopping condition with respect to the computing time, and also exhibit similar classifier performance with highly reduced size of feature subset among other evolutionary and popular approaches.

Although many studies focused their attention on discriminative power and redundancy removal of features, most of them neglect the stability of the selected features. Yu et al. addressed this issue in their two studies [47, 65]. In [47], rather than relying on typical clustering algorithms, they applied kernel density estimation accompanied by an iterative mean shift procedure to find feature clusters. Subsequently, these feature clusters were evaluated according to relevance using F-statistic and are presentative feature is selected within each cluster. The same authors extended this study in [65], where consensus feature groups were identified in an ensemble learning manner and features were extracted in the same way as their first study. The experiments conducted in both studies showed the stability of the selected features.

All the works mentioned until now are considered global FS, i.e., finding a reduced subset of global features for the entire population. However, there are cases where these

approaches are not applicable. For instance, take an image recognition task, where feature importance may alter since a set of relevant features may be important for identifying a specific object but insignificant for another object at a different position. This gap paved the way for a different technique, called Instance-wise FS that associates each feature's relationship to its labels by assigning a different selector for each instance. Interested readers to grouping and selection of features in this approach can refer to [66-67].

FS approaches based on grouping are not necessarily in the manner of grouping features into clusters and choosing representatives. Distinctly, selection of the features may happen with different cluster configurations. Moslehi et al. [68] initially implement K-means for clustering all samples for a given dataset and a sample from each cluster is chosen at random to acquire the samples with the greatest differences for the preliminary dataset. Subsequently, variances of all features on the determined samples are calculated and a predefined number of features with the highest variances are selected, thereby forming the primary dataset. Thereafter, remaining features are added gradually to this dataset and K-means clustering with a predefined number of clusters is applied iteratively in each step. Features causing changes in the structure of clusters are observed in a repetitive manner and considered significant. Other features that don't lead to any alteration in clusters are eliminated. Support Vector Machine - Recursive Cluster Elimination (SVM-RCE), proposed by Yousef et al. [69] introduced the *recursive cluster elimination* term into the community and their approach was later adopted in many studies. Since this approach was widely employed by different studies, we elaborate this method in detail by reviewing its application areas and modified usages in Section 1.2.4.

1.2.2.2 Grouping-based feature selection under unsupervised setting

As with the traditional methods in FS, many of feature grouping-based FS approaches belong to the supervised learning paradigm. Unsupervised FS is more challenging than supervised FS because of no prior knowledge about class labels and unknown number of clusters. Unsupervised FS methods typically involve (i) maximization of clustering performance by some index or (ii) selection of features based on dependency. Since this article is about FS, first one is out of scope for this study. Many statistical dependency/distance measures are available in the literature including correlation coefficient, least square regression error, Euclidean distance, entropy, and variance. Selected features in unsupervised FS methods can be evaluated in terms of both classification performance and clustering performance.

Mitra et al. [70] proposed an unsupervised FS algorithm using feature similarity. A new similarity measure called maximum information compression index is introduced in their study. Also, they demonstrated use of representation entropy for measuring redundancy and information loss quantitatively. Features are partitioned into clusters using K-nearest neighbors (KNN) principle along with a similarity measure. Entropy metric is chosen as the FS criterion and applied to select a single feature from each cluster to constitute the reduced subset. To evaluate the effectiveness of selected features, the proposed method is compared with KNN, naive Bayes and class separability including Relief-F for classification capability, and with entropy and fuzzy feature evaluation index for clustering performance. Their algorithm is rapid since no search is required and hence their study is one of the state of the art work in the literature.

Another example is the study of Li et al. [71], which uses the same similarity measure in [70] and employs a distance function to obtain clusters of features. A representative feature, having the shortest distance to others within a cluster, is selected from each cluster. Their approach is based on hierarchical clustering which enables them to choose feature subsets with different sizes by choosing from top clusters in the hierarchy. Their algorithm works for both unsupervised and supervised learning tasks. Moreover, they run clustering just one time in their algorithm. The authors presented their experimental results for both clustering and classification.

As stated previously, FS methods developed under unsupervised framework do not utilize class labels. As an example, Covões et al. [72] presents a comparative study of their approach with the algorithm proposed by Mitra et al. [70]. Again, maximal information compression index is utilized to find clusters of features. Hereafter, they employed the simplified silhouette criterion to find optimum clusters, allowing to find the number of clusters as well. The computation for simplified silhouette depends only on obtained partitions, and it is not dependent on any clustering algorithm. Hence, this silhouette not only determines the number of clusters automatically, but also it is capable of evaluating partitions acquired by any clustering algorithms. They employed the k-medoids algorithm along with the silhouette method in order to achieve optimum clusters. Then the corresponding medoid for each cluster is selected as the representative feature. The prerequisite for number of clusters known a priori in this algorithm has been overcome by the simplified silhouette since one can implement this algorithm for different values of number of clusters, and then select the best clustering according to the maximum value obtained in the silhouette.

Another study under unsupervised framework is suggested in [73], where maximal information coefficient and affinity propagation (MICAP) are exploited for selection of features. Features are chosen as the centroid of each cluster in the final step. Although they present competitive results in classification with typical classifiers, no comparison is made for clustering.

FS methods developed under supervised framework can be an inspiration to unsupervised studies. For instance, Zhou et al. [74] developed an attribute clustering algorithm along with an FS method in an unsupervised manner. They test their algorithm considering different FS methods with different classifiers and achieve slightly improved mean accuracies. The unsupervised FS algorithm proposed by Zhu et al. [75] groups features according to their SU similarities. In their clustering approach, cluster centers are firstly determined and the features are assigned to these centers subsequently. Then, the feature with the highest SU on average is selected from each cluster as a representative based on the following formula

$$AR(f, C) = \frac{\sum_{i=1}^{|C|} SU(f, f_i)}{|C|} \quad (1.14)$$

where $AR(f, C)$ is the average redundancy for a feature f in cluster C and $f_i \in C$. Their experiments showed that compared to other methods, the proposed algorithm performs more efficiently in terms of running time and in terms of the size of the reduced subset of features. Also, clustering performance of their algorithm surpasses the compared techniques for various clustering performance measurements. Apart from this, a recent hybrid work which is a combination of grouping and binary ant system (BAS) can be found in [76].

More recently, Yuan et al. [77] formulated this phenomenon as an optimization problem, where their optimization benefits from feature grouping and orthogonal constraints. Clustering performance of their algorithm shows better performance in general compared to other unsupervised FS methods.

1.2.2.3 Grouping-based feature selection under semi-supervised setting

There are cases when a significant amount of data is unlabeled and only few samples are labeled. In such a case, the learning problem is denominated as semi-supervised. Quinzán et al. [78] conducted a grouping-based FS study under this setting. In their study, the distance measure between each pair of features is computed by both conditional entropy and conditional mutual information. Next, hierarchical clustering is

applied to attain feature clusters and the feature with the highest MI is selected as the representative inside each cluster. They test the performance of their algorithm for a different number of labeled samples with other algorithms and their results exhibit satisfactory performance when there is not enough labeled data. Semi-supervised FS techniques are common in the literature and reviewed in many studies [79-81].

1.2.3 Grouping features with biological domain knowledge

The aforementioned FS approaches typically apply statistical analysis and run computational algorithms to create the feature groups. Hence, these approaches are fully data-driven and they generate the groups of features without using any domain knowledge. However, in some fields, the automatic transformation of data into information via exploiting the background knowledge in the domain is very beneficial. Background knowledge refers to the domain knowledge obtained from the literature, domain experts or from available knowledge repositories [82]. In such fields, the integration of domain knowledge into the feature selection process might improve performance, and also might reveal novel knowledge. For example, in the field of bioinformatics and computational biology, the integration of biological domain knowledge is used to improve the process of feature selection (i.e. gene selection in gene expression data analysis, in other words biomarker discovery) [83-84].

This section deals with how feature groups are created and how FS is realized using biological external sources. The main idea behind the integration of biological knowledge to FS is to apply a biological function to create groups of features (i.e., groups of genes) and then employ a learning algorithm to score these generated groups. Finally, the genes in the top scoring groups form the reduced subset of features. We would like to note that this section is especially designed for researchers working in the field of molecular biology, genetics, bioinformatics; and we believe that this section is especially informative for those with a biological background. Still, scientists working in different fields can get inspiration from the studies presented in this section and apply similar domain knowledge-based feature grouping in their problems. For example, in the field of text mining, a related tool named TextNetTopics [85] uses Latent Dirichlet Allocation (LDA) to detect topics of words, which serve as groups of features.

As one of the pioneers in this field, Bellazzi et al. [82] discussed the shift of gene expression data analysis approaches from purely data-centric approaches to integrative approaches which aim at complementing statistical analysis with knowledge acquired

from diverse available resources. The authors reported that with the growing number of knowledge bases, the field has shifted from purely data-oriented methods to methods that aim to include additional knowledge in the data analysis process [82]. The authors presented the modifications of clustering algorithms for embedding background knowledge. More specifically, the authors provide a survey of approaches that adapt distance-based, model-based and template-based clustering methods so that they take the additional background knowledge into account.

Yet as another review article in this field, recently Perscheid [83] published a survey on prior knowledge-based approaches for biomarker detection through the analysis of gene expression datasets. In that article, she evaluated the main characteristics of different integrative gene selection approaches; and she presented an overview of external knowledge bases that are utilized in these approaches [83]. It is reported that Gene Ontology (GO) [86] and Kyoto Encyclopedia of Genes and Genomes (KEGG) [87] resources are predominantly used as external knowledge bases for integrative gene selection. The author classified existing integrative gene selection approaches into three distinct categories (i.e., modifying approaches, combining approaches, module extraction approaches). The same review article presented a qualitative comparison of existing approaches and discussed the current challenges for applying integrative gene selection in practice via pointing out directions for future research. An interested reader can refer to [83] for further details.

As one of the biological knowledge-based feature grouping approaches, Support Vector Machine - Recursive Network Elimination (SVM-RNE) [88], was proposed as a variant of SVM-RCE. In [88], genes are grouped into clusters using Gene eXpression Network Analysis (GXNA) [89] and clusters with low scores are eliminated in each iteration. The algorithm terminates when some predefined constraints on the number of groups are met.

As another biological knowledge-based integrative approach, Qi and Tang attempt to incorporate GO annotations into the gene selection process, where they start by finding a discriminative score for each gene (i.e., feature) via applying IG, and eliminating those with a score of zero [90]. The next step is to annotate these genes with GO terms. After that, the score of each term is calculated as the mean of discriminative scores of associated genes involved in the respective term. The GO term with the highest score is determined and the most discriminative associated gene is selected and extracted. The steps including calculation of scores for GO terms and selection of the next most informative gene is

repeated until the final subset is formed. Their comparative results with only using IG shows the effectiveness of GO integration in the gene selection process [90]. Some other approaches for biological data integration include Bayesian methods, tree-based and network-based techniques [91].

Incorporating biological knowledge in the clustering algorithm is reported as a very challenging task [83]. Along this line, the GOstats package [92] allows one to define semantic similarity between the genes via incorporating the GO. As another example of domain knowledge-based gene selection, in SoFoCles [93], genes are initially ranked by typical filter methods such as IG, Relief-F or χ^2 , and then a reduced subset of genes is created using a predefined threshold. Next, for each gene in the reduced subset, semantically similar genes from GO are determined. Finally, top semantically similar genes are selected to enrich the reduced subset. Experimental works conducted using SoFoCles reveal enhancement in classification results by integrating biological knowledge into gene selection.

An additional study by Mitra et al. [94] adopted the Clustering Large Applications based on RAN-domized Search (CLARANS) technique to gene (i.e., feature) clustering via utilizing GOanalysis. In [94], the final reduced feature subset is composed of the genes which were medoids of biologically enriched clusters. Their experimental results showed that the incorporation of biological knowledge enhanced classifier performance and reduced computational complexity. The same authors subsequently made use of a fuzzy technique, Fuzzy Clustering Large Applications based on RAN-domized Search (FCLARANS), to obtain clusters and they selected representative genes from clusters based on the fold change [95].

The study suggested by Fang et al. [96] utilizes both KEGG and GO terms with IG. In [96], IG is applied on the initial dataset as filtering and then GO and KEGG annotations are explored for the remaining genes. As the next step, association mining is applied to this annotation information and the interestingness of the frequent itemsets is determined by averaging the original discriminative scores of the involved genes. The final gene set is attained via the selection of the highest ranked genes from the top n frequent itemsets. They assessed their method using GO, KEGG, and both against IG and study of [90]. Despite the lower rate of improvement in the overall accuracy, they are able to achieve the increase in accuracy with a significant reduction in the number of genes.

Yet as another domain knowledge-based gene selection approach, Raghu et al. [97] utilize the KEGG [87], DisGeNET [98] and other genetic meta information in their

integrated approach. In their framework, two metrics, i.e., gene importance and gene distance, are computed. Importance score for each gene is calculated using DisGeNET, which is a public platform containing gene collections associated with diseases. Distance between genes is computed based on their chromosomal locations and associations to the same diseases. Both scores are then employed to compose gene sets with maximum relevance and diversity. Compared to variance-based techniques, their method performs better in the predictive modeling task on a small scale.

Another related study developed maTE tool [99], where gene groups are created based on the miRNA-target gene information, and then each group is ordered by cross-validation. The average accuracy after a specific number of iterations determines the rank of each cluster. Genes on the top m groups are selected as the reduced subset [99].

As another example, the Grouping-Scoring-Modeling (G-S-M) [100] method benefits from the biological knowledge for its grouping step, followed by the ranking and classification steps [84]. Following the G-S-M approach, CogNet framework [101] initially implements pathfindR [102] to group the genes. The genes in each group are actually the genes of an enriched KEGG pathway, identified as a result of the active subnetwork search and functional enrichment steps of pathFindR. Then, a new dataset involving genes for the specific pathway is created for each group (i.e., pathway). These datasets are scored through Monte Carlo cross-validation (MCCV) and the pathways are ranked according to the assigned scores. Ultimately, genes found in top chosen pathways are taken as selected features and they are used for classification. Another study, developed the miRcorrNet tool [103], which finds gene groups on the basis of their correlation to miRNA expression. Afterwards, these groups are subject to a ranking function for classification. The results showed area under curve (AUC) scores above 95%, proving that miRcorrNet is capable of prioritizing pan-cancer-regulating high-confidence miRNAs. The G-S-M approach has been used by other bioinformatics tools. An example of such tools are: miRModuleNet [104], which detects groups via calculating the correlations between the mRNA and miRNA expression profiles; Integrating of Gene Ontology [105] that uses Gene Ontology information for grouping; PriPath [106] that uses KEGG pathways for grouping; GediNet [107] that uses disease gene associations as groups; 3Mint [108] that employs mRNA expression, miRNA expression and methylation profiles for grouping; and miRdisNET [109] that uses miRNA target gene information while creating the groups.

Very recently Zhang et al. [110] proposed a method called Distance Correlation

Gain-Network (DCG-Net); where they quantify distance correlation gain between features to construct the biological network. In their algorithm, a greedy search method is applied to detect network modules. The edge with the highest weight is selected, then this edge is extended with respect to correlation metric to obtain the module in the network. This is done iteratively to extract modules and the module with the highest distance correlation is selected for analysis. Their experimental results showed effective results in terms of FS and classification accuracy.

Perscheid et al. [111] comparatively evaluated traditional gene selection methods with knowledge-based methods. Their approach produces gene rankings by integrating knowledge bases and each of these rankings are evaluated with a predefined number of selected genes. Finally, the ranking with the best performance is selected. Moreover, they proposed a framework allowing external knowledge utilization, gene selection and evaluation in an automatic fashion. Although the framework seems to be knowledge base dependent, their experimental results demonstrate that incorporating biological knowledge into the gene selection process improves classification performance, decreases computational running time, and enhances the stability of selected genes.

1.2.4 Feature grouping with recursive cluster elimination

SVM-RCE, proposed by Yousef et al. [69], introduced the term *recursive cluster elimination* into the literature, and this approach predominated over SVM-RFE [25], which was widely accepted as an effective approach in the field. The superiority of SVM-RCE stems from the consideration of feature (i.e., gene) clusters instead of individual features in the classification task. In the original framework [69], the samples are initially divided into training and test sets. Then, using the training set, features are grouped into clusters. Next, feature clusters are scored according to their predictive ability using linear SVM. A predefined fraction of clusters with the lowest scores are eliminated. In each elimination step, the remaining features are fed into a classifier to quantify classification accuracy in the test set. This way, the performances of different numbers of clusters are determined as the algorithm proceeds [112]. All the steps from clustering to performance assessment are iterated and the algorithm stops when the desired number of clusters remain. The schematic diagram of this approach is illustrated in Figure 1.4.

With the advent of SVM-RCE, similar approaches emerged to perform cluster selection rather than individual feature selection. Tang et al. [113] group features into a fixed number of clusters and eliminate lower-ranked features in clusters. They repeat

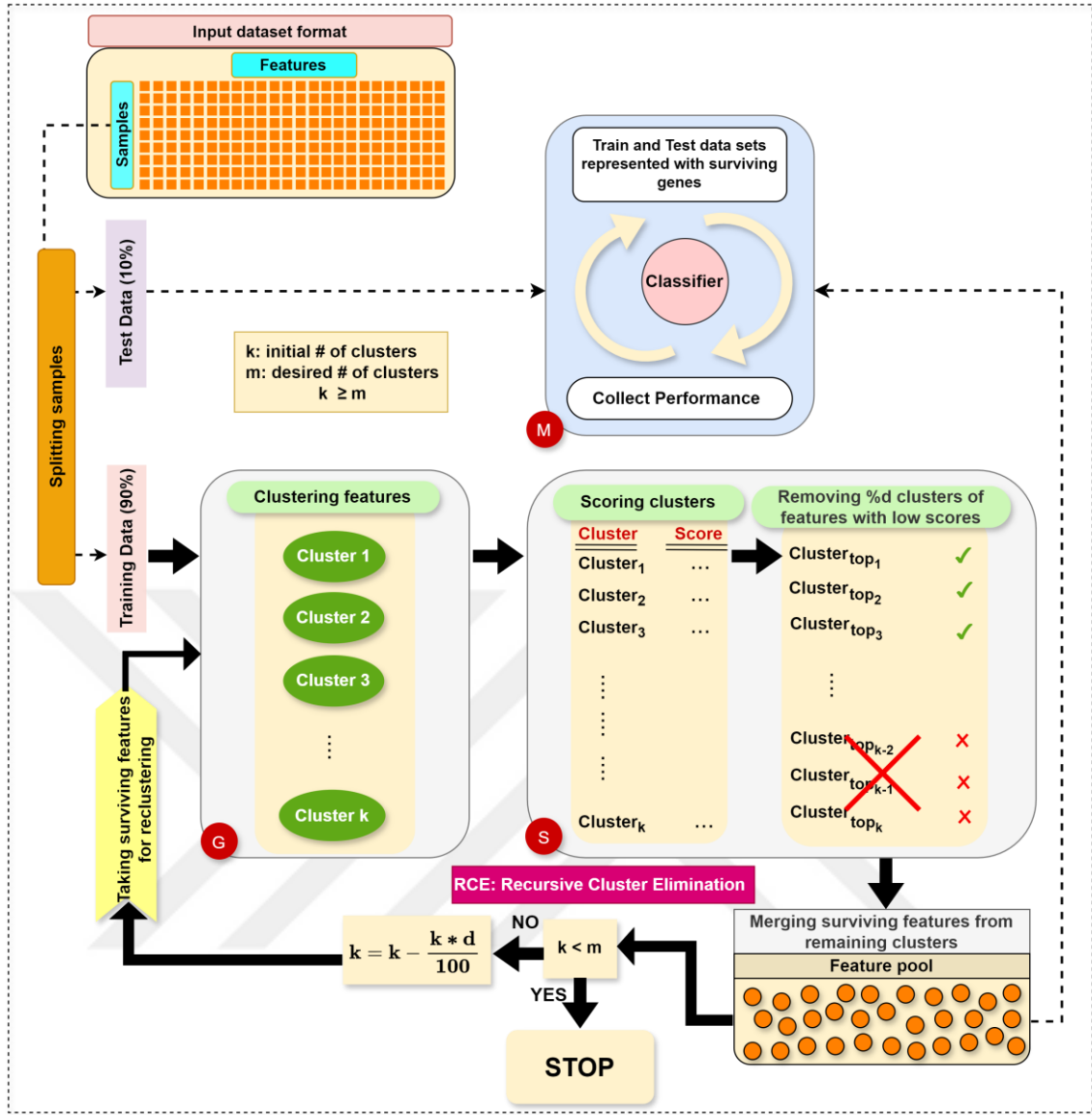


Figure 1.4 The workflow of the SVM-RCE algorithm. The grouping step (G) for grouping genes into clusters, the scoring step (S) for assigning score for each cluster and selecting significant clusters, the modeling step (M) for training the model with top-ranked clusters.

these steps until the remaining features are lowered to a predefined threshold. Du et al. [114] proposed Multi-stage, where features are first grouped into clusters, the clusters are ranked by backward elimination using SVM RFE, and those with a single feature are discarded to a certain extent. Subsequently, the features in each cluster are ranked by SVM-RFE, including clusters whose rank is higher than that cluster, and a certain proportion of features is removed. Finally, all remaining features from the clusters are gathered, and SVM-RFE is employed once again to create the final feature subset. Huang presented RBF-RCE [115], utilizing RBF [116] to calculate feature importance. The

features are then divided into clusters, and the highest importance value among the features in a cluster is assigned as the score of that cluster. RCE is then applied to remove clusters until a pre-specified number of clusters remains. Lastly, feature removal is carried out in the remaining clusters based on feature importance.

SVM-RCE, as originally described or with some variations, was used in numerous studies, including those related to neuroimaging [117-119]. Weis et al. [120] modified SVM-RCE by applying SVM on all clusters and dropping clusters one at a time. In their approach, the cluster whose removal maximizes accuracy is excluded, and this process is repeated to obtain a subset of clusters with high predictive power. To reduce execution time, Luo et al. [121] trained SVM using all clusters together and selected the highest absolute SVM feature weight within a cluster as a cluster score rather than relying on cross-validation. Their results show considerable reduction in computation time while exhibiting comparative performance as SVM-RCE. In the study associated with military service members, in addition to the statistical significance test, SVM-RCE is used to classify individuals between posttraumatic stress disorder (PTSD), postconcussion syndrome (PCS) + PTSD, and controls [122]. In their study, the features refer to the connectivity paths acquired from 125 brain regions. In their experimental works using SVM-RCE, they conclude that higher classification rate (by 4%) is achieved through imaging-based grouping than conventional grouping. Furthermore, imaging measures dominate non-imaging measures by 9% for both conventional and imaging-based groupings. Jin et al. [123] conducted a similar study and adopted a modified version of SVM-RCE in their study of brain connectivity. In their study, the diagnostic label of a novel subject is tested whether it belongs to subjects with PTSD or healthy group. The connectivity features are measured from mean resting-state time series taken from 190 regions across the entire brain. They employ SVM-RCE in their experimental work to suggest that dynamic functional and effective connectivity gives higher classification results compared to their static counterparts. Zhao et al. [124] compared SVM-RCE with other tools to detect expression profiles for identifying microRNAs associated with venous metastasis in hepatocellular carcinoma. Chaitra et al. [125] conducted a study to identify biomarkers of autism spectrum disorder (ASD) using imaging datasets. They utilized SVM-RCE to assess the classification performance for three distinct feature sets consisting of connectivity features alone, complex network (i.e., graph) measures alone, and a feature set including both. Their accuracy results are not competitive; however, the emphasis is on assessing different feature sets, especially on the combined feature set.

Furthermore, the merit of the original SVM-RCE has contributed to the development of an approach called G-S-M [100] that integrates biological prior knowledge. The G-S-M approach forms the basis for developing tools such as maTE [99], PriPath [106], GediNET [107], miRcorrNet [103], 3Mint [108], GeNetOntology [126], TextNetTopics [85], TextNetTopics Pro [127], microBiomeGSM [128], miRGediNET [129], miRdisNET [109], miRModuleNet [104], CogNet [101], and AMP-GSM [130], which integrate biological networks and prior knowledge to provide a comprehensive understanding of genetic interactions.



Chapter 2

Effect of Different Clustering Algorithms on SVM-RCE

Support Vector Machine – Recursive Cluster Elimination (SVM-RCE) is one of the several algorithms that have been developed for FS in high dimensional data. SVM-RCE involves a clustering step which is originally K-means. In this study, using various performance metrics, three alternative clustering algorithms are evaluated in this context; K-medoids, Hierarchical Clustering (HC), and Gaussian Mixture Model (GMM). This study can contribute to the development of SVM-RCE with different variations, leading to decrease in the number of selected genes, and an increase in prediction performance.

2.1 Clustering Algorithms

SVM-RCE is an impressive technique in FS because it eliminates weakly scoring clusters and combines features from surviving clusters iteratively so that highly scoring clusters including strongly relevant features remain in the final outcome. After separation of data, SVM-RCE starts with clustering whose result plays a crucial role before eliminating clusters. The clustering quality of SVM-RCE affects the eliminated clusters, thereby altering measured accuracies, average number of genes, and even running time. We considered four different clustering methods independently: K-means, k-medoids, HC, and GMM. K-means is the default algorithm used in SVM-RCE. Hence, we actually compared the original SVM-RCE with the effects of three different clustering algorithms.

2.1.1 K-means algorithm

K-means clustering [131] is one of the widely used unsupervised learning algorithms that is mainly employed to discover data structure and generate clusters. With the number of clusters (k) known in advance, the initial step is to determine cluster centers by selecting k features randomly as initial cluster centers. Then, features are assigned to these k clusters according to their proximities. When all features are assigned, cluster

centers are recalculated and features are partitioned into clusters again by their distance to new cluster centers. This phenomenon continues until there is no assignment of any feature to a new cluster. Euclidean distance is generally used to determine the nearest distance between the data point and cluster center, and defined for two vectors $x = (x_1, x_2, \dots, x_k)$ and $y = (y_1, y_2, \dots, y_k)$ as

$$d_{xy} = \left[\sum_{i=1}^k (x_i - y_i)^2 \right]^{1/2} \quad (2.1)$$

Ultimately, this algorithm aims at minimizing the objective function

$$E = \sum_{i=1}^m \sum_{j=1}^n I(x_i \in C_j) \|x_i - m_j\|^2 \quad (2.2)$$

where m and n represent total number of samples and clusters, respectively along with each sample x_i and cluster center m_j . $I(F) = 1$ if F is true and 0 otherwise.

2.1.2 K-medoids clustering

K-means clustering is sensitive to outliers since distribution of data may be deformed to a great extent by an object with a very large value. This requires another approach that is more robust to dirty and noisy data, which are likely to occur in real environments. K-medoids clustering is more robust than k-means since the k-medoids algorithm selects data points as cluster centers instead of taking the mean of object values [132]. Hence, a medoid refers to an object of a dataset. A medoid can be interpreted as the object of a cluster, whose average dissimilarity to all other objects in the cluster is the least. In other words, it's the most centrally located data point in the dataset.

K-medoids clustering is similar to k-means clustering in that it starts with randomly selecting k representative objects or medoids and each remaining object is assigned to its nearest medoid. However, k-medoids clustering employs medoids as reference points rather than averaging object values in each cluster. After the selection of initial medoids, the method updates medoids iteratively in every assignment of a data point to a cluster. This cycle continues until there is no change in any medoids. This method operates based on minimizing the sum of the dissimilarities between each data point and its corresponding medoid. An absolute error criterion is considered and defined as:

$$E = \sum_{j=1}^k \sum_{x \in C_j} |x - m_j| \quad (2.3)$$

where E is the sum of absolute error for all data points in the dataset; x is the data point representing an object in cluster C_j ; and m_j is the medoid of cluster C_j .

2.1.3 Hierarchical clustering

Hierarchical Clustering (HC) is another type of widely used clustering algorithm in data exploratory analysis. In HC, each instance of data is initially considered as a single cluster and then two clusters with the closest distance or the highest similarity are merged according to a given metric. This is repeated until all the instances belong to a single hierarchically connected cluster. The result of HC is represented as a cluster tree called dendrogram that is a graphical representation showing agglomerated clusters at each step [133]. Dendrograms allow visualization of formation of clusters consecutively and understanding the underlying structure of the data. In addition, specification for the initial number of clusters is not needed and the desired number of clusters can be obtained by selecting a cutoff point in the hierarchy.

HC approaches can be separated into two types: agglomerative (bottom-up) and divisive (top-down) [134]. Agglomerative approach represents each instance as a cluster at first and combines them iteratively until the formation of the single (root) cluster. In the divisive approach, all instances are in the same cluster at the beginning and then the cluster is divided recursively into clusters until a single instance remains in them. Agglomerative Hierarchical Clustering (AHC) is more widely used than Divisive Hierarchical Clustering (DHC) due to the sophistication of DHC, and easy implementation of AHC.

In HC, the distance between pairs of instances is determined by different distance metrics such as Euclidean, Minkowski, or Manhattan, where Euclidean distance is typically preferred. With determination of pairwise distances, we need some procedure called linkage in order to quantify the pairwise distances between clusters and merge them. Some of the most widely used linkage types are single linkage, average linkage, complete linkage, centroid linkage and Ward's method [135].

2.1.4 Gaussian Mixture Model clustering

Gaussian Mixture Model (GMM) clustering is a model-based algorithm and

employs a mixture of Gaussian distributions to train the data and implement a soft partition [136]. A GMM can be viewed as a linear association of distinct Gaussian components with density function

$$p(\mathcal{X}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathcal{X}|\mu_k, \Sigma_k) \quad (2.4)$$

where K is the number of components in the model, π_k is the mixing proportion of the k th component and

$$\mathcal{N}(\mathcal{X}|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{D/2} \sqrt{|\Sigma_k|}} \exp\left(-\frac{1}{2}(\mathcal{X} - \mu_k)^T \Sigma_k^{-1} (\mathcal{X} - \mu_k)\right) \quad (2.5)$$

is the k th Gaussian density function with mean μ_k , covariance Σ_k . The parameters of the model are represented by $\theta = \{\pi_1, \mu_1, \Sigma_1, \dots, \pi_K\}$.

Given a set of N observations $\{x_1, x_2, \dots, x_N\}$, θ is approximated by maximum likelihood estimation (MLE) method and computed as

$$\mathcal{L}(\theta) = \log p(\mathcal{X}|\theta) = \log \prod_{i=1}^N p(x_i|\theta) \quad (2.6)$$

$$= \sum_{i=1}^N \log \left(\sum_{k=1}^K \pi_k \mathcal{N}(x_i|\mu, \Sigma) \right) \quad (2.7)$$

2.2 Experimental Setup

We carried out our experiments on five gene expression datasets that are associated with different kinds of complex diseases, and these datasets were downloaded from the GEO [137] database. Each dataset has two classes, i.e., positive and negative, with a specific number of samples. Table 2.1 lists the details of the employed datasets.

We employed Knime [138] platform for execution of SVM-RCE with different clustering algorithms. We have run SVM-RCE with each clustering algorithm separately, and collected results for 10 iterations. Performance metrics using MCCV were collected for 5 repetitions. We used a fixed number of cluster pattern as 90, 80, 70, 60, 50, 40, 30, 20, 10, 5, 2, 1 during cluster elimination at each iteration. For each cluster level, we obtained gene number and other performance metrics and got averaged values for 10 iterations. In order to obtain a balanced data, if the number of one class label is more than

Table 2.1 Basic information on gene expression datasets

Dataset	# of Genes	# of Samples	Classes
GDS2547	12647	164	positive: 89 negative: 75
GDS3268	44290	200	positive: 129 negative: 71
GDS3646	22186	132	positive: 110 negative: 22
GDS3875	22646	117	positive: 93 negative: 24
GDS5037	41001	108	positive: 88 negative: 20

twice as large as another, then the size of the bigger class is reduced to twice the smaller label. If not, then the class with the bigger size is reduced, and it is equalized to the smaller one.

2.3 Results

In our experiments, we examined the average number of features (genes) and AUC metrics obtained for four clustering algorithms. In order to make a fair comparison, we have selected a similar amount of average number of genes along with their corresponding average AUC values. Table 2.2 shows the average gene number and AUC for 10 iterations. Columns A, B, C and D represent use of K-means, K-medoids, HC and GMM, respectively.

Table 2.2 Average gene and AUC values for four clustering algorithms

Dataset	Avg gene #				Avg AUC #			
	K-means	K-medoids	HC	GMM	K-means	K-medoids	HC	GMM
GDS2547	72.1	60.2	76.7	72.8	0.80	0.85	0.83	0.74
GDS3268	41.5	51.5	63.3	43.8	0.70	0.61	0.73	0.72
GDS3646	27.8	22.5	21.8	24.7	0.76	0.56	0.73	0.66
GDS3875	42.1	39.8	28.1	45.4	0.79	0.73	0.77	0.77
GDS5037	38.3	27.0	31.2	37.5	0.72	0.56	0.56	0.63
Average:	44.4	40.2	44.2	44.8	0.75	0.66	0.72	0.70

K-medoids clustering obtains the highest accuracy only for GDS2547 and HC provides almost the same accuracy with more genes. In other datasets, k-medoids is not so effective and gives the lowest classification performance. For GDS3268, HC and

GMM perform similarly but GMM achieves it with less number of features. K-means presents a similar performance with a similar number of genes as GMM.

In GDS3646, k-means slightly outperforms HC with a few more genes. For GDS3875, k-means provides the highest classification accuracy with almost the same number of genes as GMM. HC and GMM have the same performance and they reach an accuracy similar to k-means but HC does it with around less than one and a half times as many genes as k-means and GMM. For GDS5037, k-means surpasses all other clustering algorithms significantly with almost the same number of genes as GMM. Overall, SVM-RCE with k-means, which is the original case, is superior to integration of other clustering algorithms. Note that HC provides a comparable accuracy with k-means with similar number of genes. It is also noteworthy that the deviation in accuracies is minimal for k-means compared to others and this shows stability of it among datasets. Although k-medoids attains the smallest number of genes on average, its effect on performance is the lowest.

We have also measured the execution time of SVM-RCE with different clustering configurations for 10 iterations. Average time consumed by each configuration per iteration is shown in Figure 2.1. As shown in Figure 2.1, k-medoids algorithm has the highest

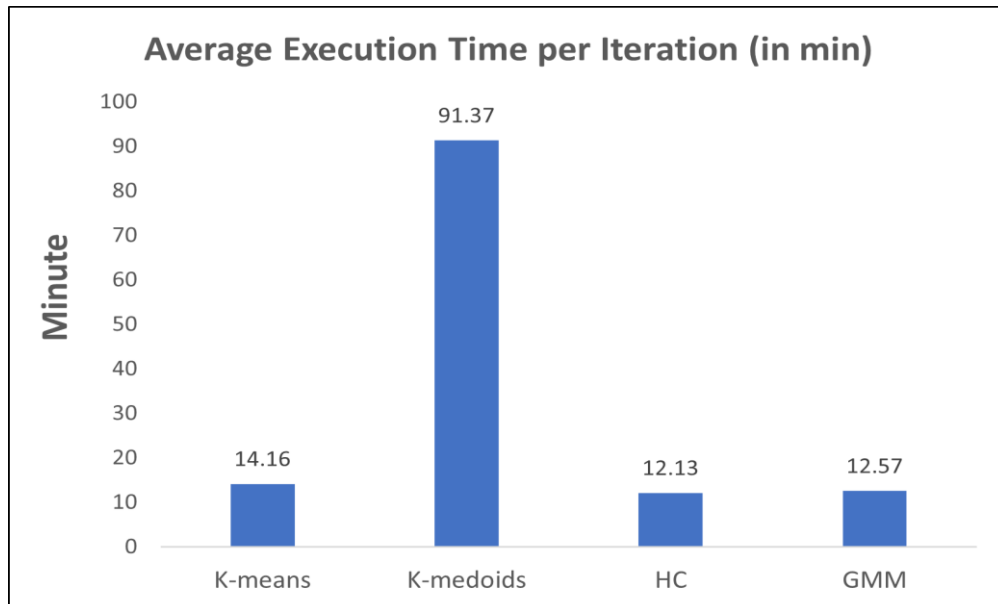


Figure 2.1 The effects of clustering algorithms on execution time per iteration

execution time per iteration besides its low performance. Actually, it is known that k-medoids algorithm can take high execution time due to its exhaustive search in

ascertaining the cost for swapping medoids [139]. HC provides the smallest execution time, and the execution times of other clustering algorithms are very close to HC.

2.4 Discussion

The aim of this study is to analyze the effects of different clustering algorithms on SVM-RCE and make comparisons between them. We considered four different clustering methods separately: k-means, k-medoids, HC, and GMM. As K-means is the default clustering algorithm in SVM-RCE, we actually compared the original SVM-RCE with the effects of three different clustering algorithms. The results of the experiments showed the dominance of k-means algorithm in classification performance, and competitive performance of HC. K-medoids presented the lowest performance in general and it achieved the best accuracy just for one dataset. The performance of GMM was close to HC, but with slightly less accuracy. It has also been observed that all clustering algorithms, except k-medoids, have similar effects on the running time.

To sum up, impacts of k-means and HC are noteworthy with respect to average accuracies. K-means is identified as the most stable algorithm across different datasets. The gap between k-means and HC in terms of accuracy is not large, making HC a promising clustering algorithm for SVM-RCE.

Chapter 3

RCE-IFE: Recursive Cluster Elimination with Intra-cluster Feature Elimination

In this study, we extend SVM-RCE by integrating feature elimination within surviving clusters at each step of cluster reduction. Through this newly added phase, less contributing features in each cluster are excluded, leading to deeper dimension reduction and also improvements in feature subset quality, classification performance, and running time. The proposed Recursive Cluster Elimination with Intra-cluster Feature Elimination (RCE-IFE) tool can tackle large-scale datasets. We assess dimensionality reduction and discriminatory capabilities of RCE-IFE on various high-dimensional datasets from different biological domains. Compared with a similar tool, Multi-stage, RCE-IFE gives a similar average accuracy rate using fewer features on four cancer-related datasets. The comparability of RCE-IFE is also verified with other biological domain knowledge-based G-S-M tools, including mirGediNET, 3Mint, and miRcorrNet. Additionally, the biological relevance of the selected features by RCE-IFE is evaluated. The proposed method also exhibits high consistency in terms of the selected features across multiple runs. Our experimental findings imply that RCE-IFE provides robust classifier performance and significantly reduces feature size while maintaining feature relevance and consistency.

3.1 Materials and Methods

3.1.1 Datasets

The datasets used in this study cover a broad range of biological domains and disease types. Table 3.1 summarizes the utilized 20 datasets on gene expression, methylation, and microRNA (miRNA) accessed from GEO [140] and TCGA [141] databases. While gene expression datasets are available at GEO; TCGA-BLCA.methylation450, TCGA-BLCA.mirna and TCGA-BRCA.methylation450 datasets

can be obtained from UCSC Xena repository accessible at (<https://xenabrowser.net/datapages/>) [142]. TCGA-BLCA.methylation450 and TCGA-BLCA.mirna datasets are accessible in the GDC TCGA Bladder Cancer (BLCA) cohort. TCGA-BRCA.methylation450 dataset can be found in the GDC TCGA Breast Cancer (BRCA) cohort.

Table 3.2 describes seven metagenomics datasets involving colorectal cancer (CRC), Inflammatory Bowel Disease (IBD), Inflammatory Bowel Disease Multi-omics Database (IBDMDB), and Type 2 diabetes (T2D). CRC_species, CRC_pathway, and CRC_enzyme datasets were created and presented as a supplementary material in Beghini et al.'s study [143]. The IBD dataset is obtained from the European Nucleotide Archive (ENA) database with accession number ERA000116. The IBDMDB dataset is available at Sequence Read Archive (SRA) with accession number PRJNA398089. The T2D dataset is provided by NCBI SRA with accession numbers SRA045646 and SRA050230. The CRC_species_II dataset can be accessed from the ENA database with accession number PRJEB6070.

Lastly, four publicly available cancer-related datasets are described in Table 3.3. The leukemia dataset can be downloaded from the link available at <https://www.openintro.org/data/index.php?data=golub>. The prostate dataset can be extracted from the R package SIS (details are available at <https://cran.r-project.org/web/packages/SIS/SIS.pdf>). The breast dataset, available in the Breast Cancer (vantVeer 2002) cohort, can be accessed through the UCSC Xena platform [142] at <https://xenabrowser.net/datapages/>. Finally, the DLBCL dataset can be downloaded at <https://github.com/ramhiser/datamicroarray>.

3.1.2 Proposed Algorithm

The proposed algorithm, called RCE-IFE, follows an iterative process in which each iteration involves the following two elimination steps: firstly, weakly-scoring clusters are removed; secondly, intra-cluster elimination is performed for the low-scoring features in the surviving clusters [144]. Let M be the $K \times 1$ vector of cluster numbers in descending order. The number of clusters in the i th iteration is M_i , where $M_i < \dots < M_K$ represents a user-supplied decreasing sequence. The user also controls the rate of intra-cluster feature elimination. The data, consisting of p samples and q features, is organized in a $p \times q$ matrix, D , accompanied by a vector of class labels S . To begin with, we create a training set and a test set by randomly splitting the samples, i.e., the rows of D , into D_{train} ($r_{\text{train}}\%$)

Table 3.1 Basic information about the gene expression, miRNA, and methylation datasets

Dataset	#samples	Label: count	#features	Disease Type
GDS1962	180	pos:157, neg:23	54613	Glioma
GDS2519	105	pos:50, neg:55	22283	Parkinson's disease
GDS2547	164	pos:75, neg:89	12646	Prostate cancer
GDS2609	22	pos:12, neg:10	54635	Colorectal cancer
GDS3268	200	pos:129, neg:71	44289	Ulcerative colitis
GDS3646	132	pos:110, neg:22	22185	Celiac disease
GDS3794	33	pos:18, neg:15	48702	Rheumatoid arthritis
GDS3837	120	pos:60, neg:60	30622	Non-small cell lung carcinoma
GDS3874	117	pos:93, neg:24	22284	Diabetes
GDS3875	117	pos:93, neg:24	22645	Diabetes
GDS3929	64	pos:45, neg:19	24527	Tobacco smoke-related defects
GDS4228	166	pos:147, neg:19	4776	Human immunodeficiency virus
GDS4824	21	pos:13, neg:8	54635	Prostate cancer
GDS5037	108	pos:88, neg:20	41000	Severe asthma
GDS5093	56	pos:47, neg:9	54613	Acute dengue
GDS5499	140	pos:99, neg:41	48803	Pulmonary hypertension
GSE157103	126	pos:100, neg:26	19472	Coronavirus disease 2019
TCGA-BLCA.methylation450	425	pos:408, neg:17	20623	Bladder cancer
TCGA-BLCA.mirna	425	pos:408, neg:17	1881	Bladder cancer
TCGA-BRCA.methylation450	124	pos:36, neg:88	15770	Breast cancer

Table 3.2 Basic information about the metagenomics datasets

Dataset	#samples	Label: count	#features	Disease Type	Reference
CRC_enzyme	1262	pos: 600, neg: 662	2875	Colorectal cancer	[143]
CRC_pathway	1262	pos: 600, neg: 662	549	Colorectal cancer	[143]
CRC_species	1262	pos: 600, neg: 662	917	Colorectal cancer	[143]
CRC_species_II	108	pos: 60, neg: 48	528	Colorectal cancer	[145]
IBD	382	pos: 148, neg: 234	534	Inflammatory bowel disease	[146]
IBDMDB	1638	pos: 1209, neg: 429	578	Inflammatory	[143]

				bowel disease	
T2D	290	pos: 155, neg: 135	587	Type 2 diabetes	[147]

Table 3.3 Basic information about the cancer-related datasets

Dataset	#samples	Label: count	#genes	Reference
Breast	117	positive: 28, negative: 89	11885	[148]
DLBCL	77	DLBCL: 58, FL: 19	7129	[149]
Leukemia	72	ALL: 47, AML: 25	7129	[150]
Prostate	102	tumor: 52, normal: 50	12600	[151]

and D_{test} ($r_{\text{test}}\%$). A two-sided t-test is applied to each feature in D_{train} to select the top 1000 features with the smallest p-values. Let F_0 denote the set of filtered features. The subsequent operations are performed recursively. In the i th iteration, $i = 1, \dots, K-1$, K-means is used to group F_{i-1} into M_i clusters $C_i = \{C_{i1}, C_{i2}, \dots, C_{iM_i}\}$. Next, each $C_{ij} \in C_i$ is assigned the mean of t estimates of classification accuracy. Algorithm 1 presents the pseudocode for assigning a score to a cluster. Clusters with the lowest $M_i - M_{i+1}$ scores are eliminated, leaving M_{i+1} clusters in C_i . Within each cluster, intra-cluster feature importance weights are estimated by Random Forest (RF) algorithm, and the lowest-ranked $f\%$ of features are eliminated. Algorithm 2 gives the pseudocode for scoring the features in a surviving cluster. Finally, D_{train} and D_{test} are updated to contain only the surviving features, and the performance of the learned model is computed using RF. To give a brief example, if $M = \{100, 90, 70\}$, in the first iteration, the algorithm eliminates $100 - 90 = 10$ clusters, performs intra-cluster feature elimination on $f\%$ of the features in the 90 surviving clusters, and outputs performance results based on 90 clusters. In the second iteration, the algorithm eliminates $90 - 70 = 20$ clusters, conducts intra-cluster feature elimination in the 70 surviving clusters, and generates performance results based on 70 clusters. The algorithm would then halt, yielding performance measures for both the 90 and 70 clusters. The full steps of RCE-IFE are given in Algorithm 3.

Algorithm 1: Cluster Scoring – Score ($C, D_{\text{train}}, s, t$)

C = a set of feature identifiers

D_{train} = training set with a certain portion of the original sample set

s = set of class labels corresponding to samples in D_{train}

D_{train_C} = subset of D_{train} containing features only in C with the corresponding sample

values

t = number of partitions

$acc = []$ an empty array of length t

Step 1: for $i = 1:t$ do

Step 2: Split the samples randomly in D_{trainC} into d_{train} (70%) and d_{test} (30%) and, correspondingly, s into s_{train} and s_{test}

Step 3: Train RF Learner using d_{train} and s_{train}

Step 4: $acc[i] =$ test RF on d_{test} and s_{test} - compute accuracy of the model

Step 5: end

Step 6: return mean (acc)

Algorithm 2: Intra-cluster Feature Scoring – Importance (C, D_{train}, s)

C = a set of feature identifiers

D_{train} = training set with a certain portion of the original sample set

s = vector of class labels corresponding to samples in D_{train}

D_{trainC} = subset of D_{train} containing features only in C with the corresponding sample values

Step 1: Train RF Learner using D_{trainC} and s .

Step 2: Get measures of feature importance.

Algorithm 3: RCE-IFE

Input: $D = p \times q$ matrix representing p samples and q features.

S is a $p \times 1$ vector of class labels. Each S_j is either pos or neg

M is a $K \times 1$ vector of numbers of clusters arranged in decreasing order

r_{train} : percentage of samples in the training set. r_{test} is defined to be $100 - r_{train}$

f = percentage of intra-cluster features to be removed

Output: The performance results for different numbers of clusters

Step 1: Partition the rows of D randomly into D_{train} (90%) and D_{test} (10%)

Step 2: Apply the t-test to compare the two classes w.r.t. each feature. Define F_0 as the set of features with the smallest 1000 p-values

Step 3: For $i = 1, 2, \dots, K-1$, do

Step 4: Cluster the features in F_{i-1} using D_{train} to create a partition, $C_i = \{C_{i1}, C_{i2}, \dots, C_{iM_i}\}$

Step 5: For each $C_{ij} \in C_i, j = 1, 2, \dots, M_i$, do

Step 6: Call Score (C_{ij} , D_{train} , s , t) to assign a score to each cluster C_{ij}

Step 7: End for

Step 8: Rank clusters in descending order and delete the clusters with the lowest $M_i - M_{i+1}$ scores. At the end of this step, the number of clusters in C_i will be M_{i+1}

Step 9: For each $C_{ij} \in C_i$, $j = 1, 2, \dots, M_{i+1}$, do

Step 10: Call Importance (C_{ij} , D_{train} , s) to compute an importance score for each feature in cluster C_{ij}

Step 11: Remove the $f\%$ features with the lowest importance score in cluster C_{ij}

Step 12: End for

Step 13: Gather the remaining features in all clusters into $F^* = \bigcup_{l=1}^{M_{i+1}} \{C_{il}\}$

Step 14: Update D_{train} and D_{test} to have the same features as F^*

Step 15: Compute performance metrics using D_{train} and D_{test}

Step 16: Define $F_{i+1} = F^*$

Step 17: End for

The main steps in RCE-IFE are: Grouping (G), Scoring (S), Intra-cluster Feature Elimination (IFE), and Modeling (M). The workflow of RCE-IFE is illustrated in Figure 3.1, and details of each step are explained next.

Grouping (G) step: The first step involves utilizing the training dataset to group the currently active genes into a predetermined number of clusters. In our tool, K-means is employed, though other methods could also be applied. Let $C = \{c_1, c_2, \dots, c_k\}$ represent the set of feature clusters. For each cluster c_j , a two-class subdataset is constructed, containing all features in c_j , along with the corresponding sample values and class labels. As a result, k two-class subdatasets are generated in this step. Figure 3.2 illustrates this process, showing a training set of ten samples (rows) and ten genes (columns). In this example, four gene clusters (upper right) are formed, leading to the generation of four two-class subdatasets, which are then fed into the S step for scoring, ranking, and elimination.

Scoring (S) step: Each subdataset is assigned a score using a supervised learning algorithm such as Support Vector Machine (SVM) or Random Forest (RF) (here, we use RF). The subdataset is randomly partitioned into training (70%) and test (30%) sets,

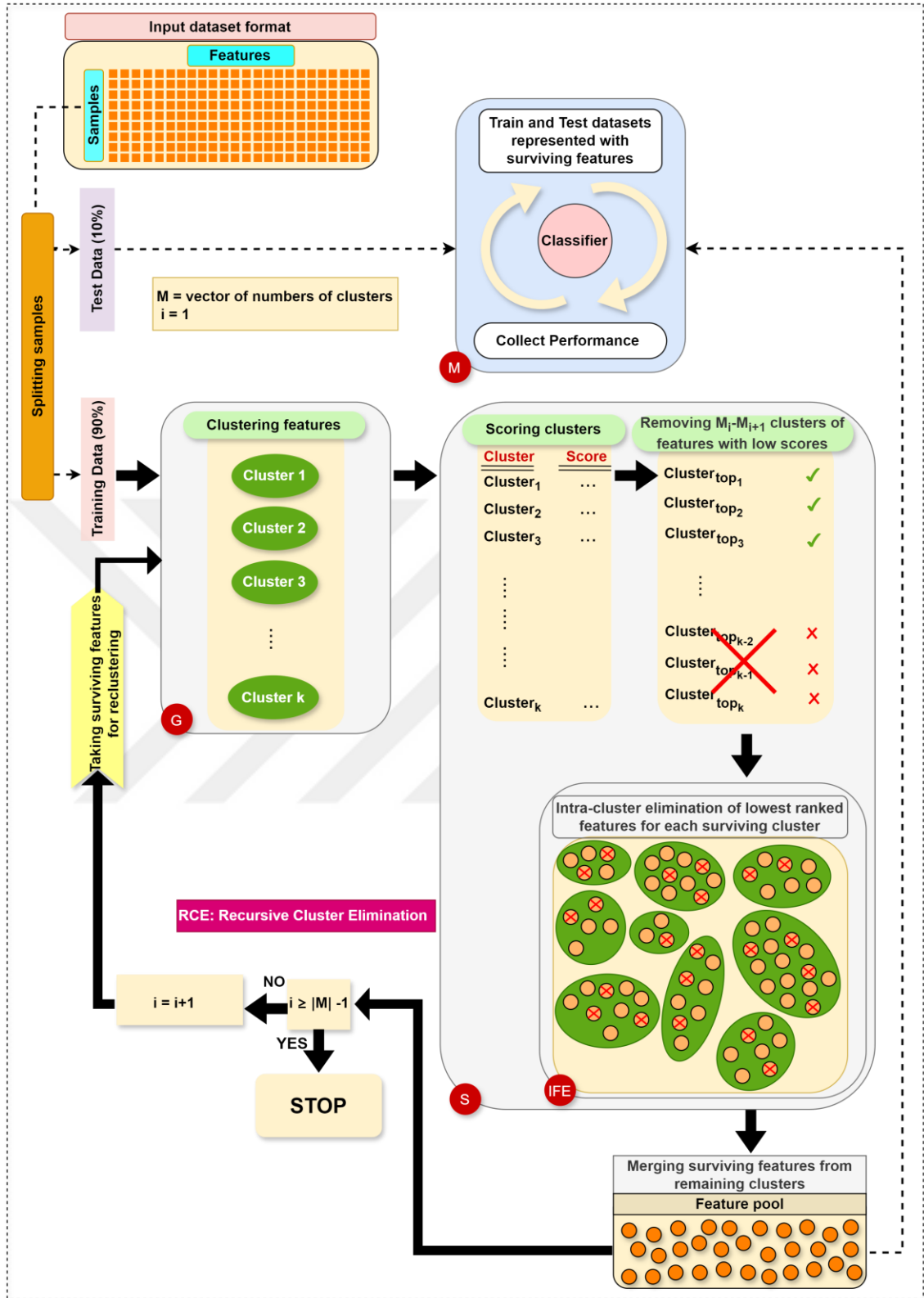


Figure 3.1 The workflow of RCE-IFE algorithm

where the training data is used for learning the model, and the test data is used for validation. This process is repeated t times and the average classification accuracy serves

as the score for the subdataset. Stratified random sampling is applied during partitioning to guarantee that the training and test sets have nearly the same proportion of class labels as the original dataset. In summary, scoring a subdataset is accomplished through the randomized stratified t-fold cross validation [152]. The scoring step is depicted in Figure 3.3. It is worth noting that any accuracy index can be used for scoring. Once subdatasets are scored, they are ranked in descending order, and $M_i - M_{i+1}$ clusters with the lowest scores are eliminated as stated before. Eliminating a subdataset means that all features in it are eliminated.

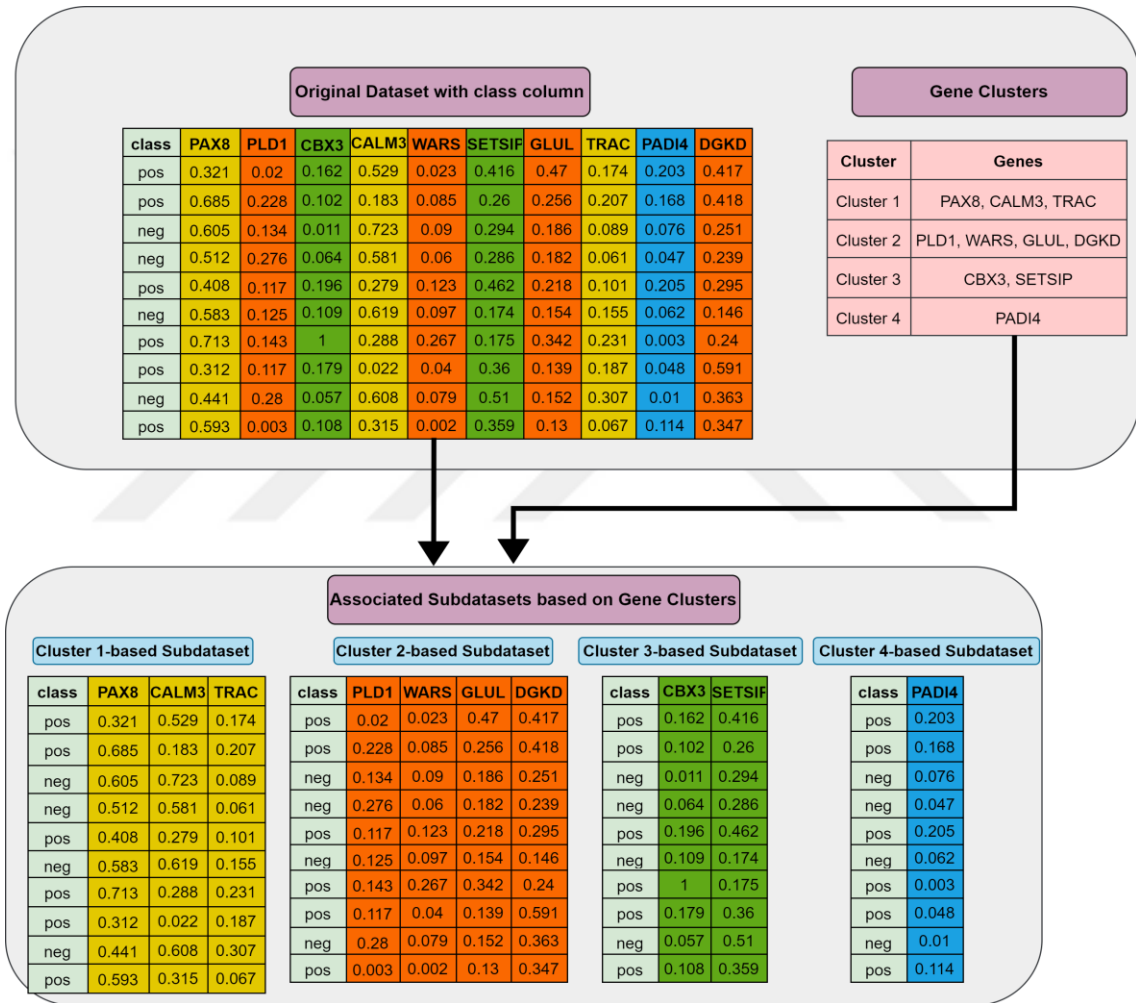


Figure 3.2 Extraction of subdatasets based on gene clusters

Internal Feature Elimination (IFE) step: Our algorithm eliminates not only low-scoring clusters but also weakly-scoring features within surviving feature clusters. Feature scores are mainly weights or coefficients estimated by a classification algorithm (in our method, RF). The rate of feature removal is specified in advance. IFE plays a crucial role by further reducing the number of features at each cluster elimination step.

This novel process facilitates the exclusion of irrelevant or redundant features, which is, in turn, expected to enhance the generalizability of the model and obtain the same (or better) performance with a smaller number of features. In addition, IFE allows the most important clusters to retain a minimum number of features at the end.

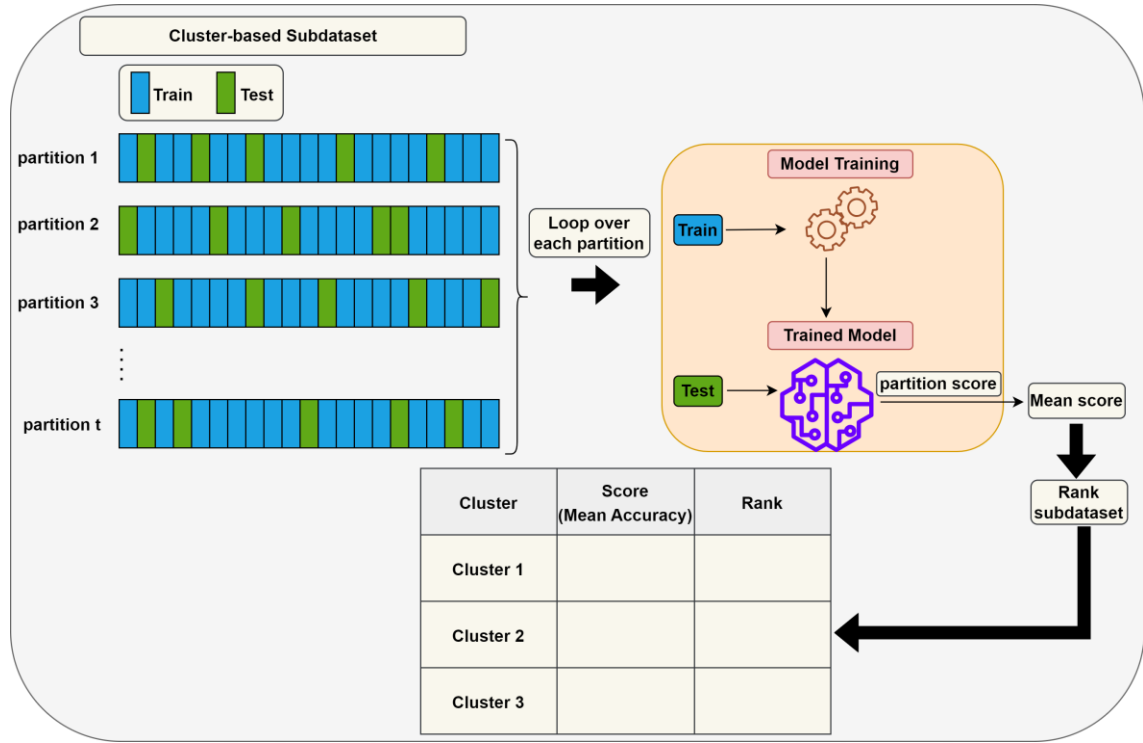


Figure 3.3 Assigning a score to a subset

Modeling (M) step: Following the removal of clusters with low scores and intra-cluster elimination of low-scoring features in the remaining clusters, the retained features from the surviving clusters are pooled. Features in this pool are then exploited to construct the training and test datasets, as explained in the G step. Subsequently, an RF model is trained on the training dataset and assessed on the test dataset. At each elimination step, i.e., for each reduced number of clusters, various performance statistics are recorded. Ultimately, we obtain performance results for multiple cluster numbers, and the algorithm terminates after gathering performance results for the final cluster number in M.

Several quantitative metrics, including Accuracy, Sensitivity, Specificity, Precision, and F-measure, were computed to evaluate the model performance using the formulations below:

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{FP} + \text{FN} + \text{TN}),$$

$$\text{Sensitivity} = \text{TP} / (\text{TP} + \text{FN})$$

$$\text{Specificity} = \text{TN} / (\text{TN} + \text{FP})$$

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

$$\text{F-measure} = 2\text{TP} / (2\text{TP} + \text{FP} + \text{FN})$$

where TP, FP, TN, and FN refer to the number of correctly predicted positive samples, incorrectly predicted negative samples, correctly predicted negative samples, and incorrectly predicted positive samples, respectively. Furthermore, AUC is a performance measurement that evaluates the ability of a model to discriminate between two classes, such as the positive class (existence of a disease) and the negative class (lack of a disease). Besides, Cohen’s kappa score is a metric that quantifies the degree of agreement between two raters and evaluates the effectiveness of the ML model [153]. All these metrics are calculated for each cluster number in our approach.

In addition to the aforementioned performance metrics, our method generates a ranked list of features in descending order that persisted successfully through the elimination process based on the number of clusters they survived.

3.2 Results

This section presents the performance of the proposed RCE-IFE approach. We conducted experiments to compare RCE-IFE with SVM-RCE using several types of datasets. We also compared RCE-IFE with some traditional FS methods using various classifiers. The proposed approach was further tested on four publicly available cancer-related datasets and compared with Multi-stage algorithm [114], a method that bears similarities to RCE-IFE. We also compared RCE-IFE with biological domain knowledge-based G-S-M tools, including mirGediNET [129], 3Mint [108], and miRcorrNet [103]; and manifested the biological relevancy of the high-scoring features selected by RCE-IFE. Lastly, the consistency of the proposed algorithm in terms of selected features is verified via comparing the selected features list among multiple runs.

3.2.1 Experimental Setup

All methods were repeated 100 times to provide stability in results. To ensure balance in datasets, we applied undersampling to remove the samples from the majority class while preserving all of the samples in the minority class. In the cluster scoring step, we applied 10-fold cross validation (i.e., $t = 10$). For RCE-IFE, we considered the results

of two clusters among various cluster numbers. The values presented in our analyses are the average results of 100 repetitions. In RCE-IFE, the rate of features to be removed from a surviving cluster is set to be 10% (i.e., $f = 10$) if the cluster contains more than five features. AUC is used to evaluate the classifier performance, while the accuracy metric is employed for comparison with Multi-stage algorithm [114], in Section 3.2.5.

We performed our analyses on Knime [138], an open-source software, due to its simplicity and support for graphical representations. In addition, Knime is an extremely integrative tool that allows the integration of different scripts in many coding languages. All experiments were conducted on an Intel Core i9-9900 with 64 GB of RAM.

3.2.2 Comparative performance evaluation of RCE-IFE with SVM-RCE on gene expression, miRNA, and methylation datasets

In this section, we compare our proposed approach, RCE-IFE, with SVM-RCE in terms of classification performance, the number of selected features, and execution time. We employed two classifiers, i.e., RF and SVM, individually in our approach. In other words, the steps in scoring clusters and features in surviving clusters were achieved independently using RF and SVM. RF is the default classifier in RCE-IFE, and we denote the approach for integration of SVM into RCE-IFE as RCE-IFE-SVM. Table 3.4 shows the results on 20 publicly available datasets. The first column corresponds to AUC values, followed by the number of features found by each technique. The last row refers to the average AUC and feature size values obtained by each algorithm across different datasets. We can observe that all methods achieve similar AUC values, where SVM-RCE is outperformed by RCE-IFE-SVM and RCE-IFE by 1% and 2%, respectively. However, a significant difference is observed in the number of selected features between SVM-RCE and RCE-IFE methods. RCE-IFE-SVM selects roughly half as many features as SVM-RCE whereas RCE-IFE picks almost two and a half times fewer features than SVM-RCE. Hence, our proposed approach significantly reduces the feature subset size while maintaining the classifier performance. Additional performance metrics can be found in Table A.1, Table A.2, and Table A.3.

Regarding running time, RCE-IFE and RCE-IFE-SVM have less time complexity than SVM-RCE, as illustrated for miRNA and methylation datasets in Figure 3.4. Note that our proposed approach might seem to have a trade-off between an additional intra-cluster feature elimination step and further feature removal. However, feature removal means dimensionality reduction and outweighs the inclusion of the intra-cluster feature

Table 3.4 Comparative performance evaluation of RCE-IFE with SVM-RCE: Mean AUC and mean size of the feature subsets on gene expression, miRNA and methylation datasets

Dataset	AUC			#Features		
	SVM-RCE	RCE-IFE-SVM	RCE-IFE	SVM-RCE	RCE-IFE-SVM	RCE-IFE
GDS1962	0.98 ± 0.05	0.99 ± 0.03	0.98 ± 0.05	76.39 ± 30.23	33.93 ± 14.64	28.52 ± 10.76
GDS2519	0.50 ± 0.18	0.56 ± 0.17	0.55 ± 0.18	46.05 ± 18.44	32.69 ± 10.58	30.99 ± 9.37
GDS2547	0.80 ± 0.11	0.83 ± 0.10	0.80 ± 0.11	109.05 ± 38.84	67.84 ± 16.57	51.19 ± 16.76
GDS2609	0.97 ± 0.14	0.97 ± 0.14	0.99 ± 0.07	59.03 ± 25.06	26.34 ± 8.96	19.99 ± 7.21
GDS3268	0.80 ± 0.12	0.78 ± 0.12	0.76 ± 0.13	99.40 ± 40.34	41.44 ± 14.29	40.52 ± 14.45
GDS3646	0.74 ± 0.23	0.77 ± 0.22	0.67 ± 0.25	38.89 ± 23.88	22.56 ± 9.70	26.44 ± 10.12
GDS3794	0.96 ± 0.12	0.94 ± 0.14	0.93 ± 0.17	63.67 ± 24.61	25.52 ± 8.08	19.25 ± 6.02
GDS3837	0.98 ± 0.03	0.98 ± 0.04	0.98 ± 0.04	124.68 ± 40.48	62.60 ± 21.21	49.82 ± 16.07
GDS3874	0.73 ± 0.20	0.75 ± 0.20	0.86 ± 0.16	42.19 ± 16.94	21.53 ± 6.32	24.94 ± 7.38
GDS3875	0.84 ± 0.15	0.86 ± 0.14	0.83 ± 0.14	41.55 ± 14.32	22.22 ± 7.14	26.26 ± 7.95
GDS3929	0.51 ± 0.28	0.48 ± 0.31	0.50 ± 0.26	17.33 ± 7.30	7.26 ± 2.42	19.82 ± 6.80
GDS4228	0.48 ± 0.26	0.46 ± 0.27	0.80 ± 0.20	12.17 ± 6.18	6.63 ± 2.68	21.20 ± 6.73
GDS4824	0.95 ± 0.22	0.95 ± 0.22	0.90 ± 0.30	61.57 ± 24.17	23.32 ± 8.39	14.38 ± 5.87
GDS5037	0.73 ± 0.24	0.72 ± 0.22	0.75 ± 0.22	37.62 ± 14.15	19.00 ± 5.55	21.99 ± 7.39
GDS5093	0.90 ± 0.22	0.91 ± 0.22	0.90 ± 0.22	42.35 ± 23.24	15.40 ± 9.09	18.15 ± 7.56
GDS5499	0.96 ± 0.07	0.95 ± 0.09	0.92 ± 0.10	179.76 ± 45.84	69.92 ± 14.77	71.34 ± 22.38
GSE157103	0.94 ± 0.10	0.97 ± 0.07	0.93 ± 0.11	95.63 ± 35.05	42.30 ± 14.18	31.76 ± 11.15
TCGA-BLCA.methylation450	0.98 ± 0.06	0.98 ± 0.05	0.97 ± 0.08	73.81 ± 28.14	37.65 ± 13.28	27.15 ± 10.57
TCGA-BLCA.mirna	0.97 ± 0.08	0.98 ± 0.06	0.97 ± 0.08	109.19 ± 107.69	49.86 ± 42.42	25.95 ± 22.90
TCGA-BRCA.methylation450	0.98 ± 0.05	0.98 ± 0.04	0.99 ± 0.03	144.27 ± 49.87	62.33 ± 21.95	53.95 ± 18.80
Average	0.83 ± 0.15	0.84 ± 0.14	0.85 ± 0.15	73.73 ± 30.74	34.52 ± 12.61	31.18 ± 11.31

elimination step in favor of shortening execution time. In Figure 3.4, we observe that RCE-IFE-SVM shows a moderate reduction in running time compared to SVM-RCE to a certain degree due to the contribution of the intra-cluster elimination step. On the other hand, RCE-IFE has the shortest running time by far. It is also noteworthy that all algorithms achieve the same AUC performance (98%) on average, with different execution times for the datasets.

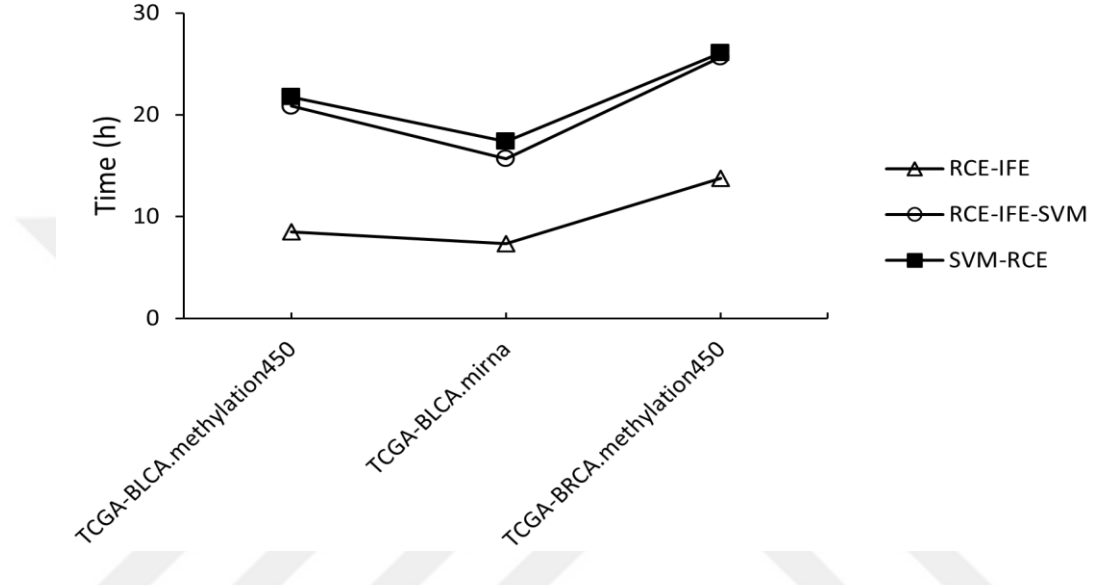


Figure 3.4 Running time comparison of RCE-IFE, RCE-IFE-SVM and SVM-RCE on miRNA and methylation datasets

3.2.3 Comparative performance evaluation of RCE-IFE with SVM-RCE on metagenomics datasets

To further highlight the dominance of RCE-IFE over SVM-RCE, we tested both methods on seven metagenomics datasets for comparison. The results in Table 3.5 show that RCE-IFE surpasses SVM-RCE for all datasets in performance and reduced feature subset. Notably, remarkable improvements are observed for CRC and IBDMDB datasets, as seen in the % columns in Table 3.5. RCE-IFE proves superior to SVM-RCE in both improving classifier performance and reducing feature size. Besides, RCE-IFE generally provides satisfactory results except for T2D, indicating that it is impressive on metagenomics data types. The superiority of RCE-IFE over SVM-RCE in terms of other performance metrics is presented in Table A.4.

Table 3.5 Comparative performance evaluation of RCE-IFE with SVM-RCE: Mean AUC and mean size of the feature subsets on metagenomics datasets

Dataset	AUC		% increase	#Features		% decrease
	SVM-RCE	RCE-IFE		SVM-RCE	RCE-IFE	
CRC_enzyme	0.70 ± 0.04	0.76 ± 0.04	8.6	78.51 ± 27.33	52.32 ± 17.91	33.4
CRC_pathway	0.68 ± 0.04	0.70 ± 0.04	2.9	83.97 ± 50.93	31.60 ± 9.60	62.4
CRC_species	0.67 ± 0.07	0.80 ± 0.04	19.4	96.24 ± 72.55	34.65 ± 8.63	64.0
CRC_species_II	0.50 ± 0.20	0.82 ± 0.14	64.0	41.50 ± 47.58	16.31 ± 5.84	60.7
IBD	0.82 ± 0.08	0.87 ± 0.06	6.1	89.57 ± 71.94	22.12 ± 5.75	75.3
IBDMDB	0.77 ± 0.06	0.93 ± 0.04	20.8	100.07 ± 66.79	30.53 ± 8.38	69.5
T2D	0.59 ± 0.12	0.66 ± 0.10	11.9	63.50 ± 58.06	16.23 ± 4.13	74.4
Average	0.68 ± 0.09	0.79 ± 0.07	-	79.05 ± 56.45	29.11 ± 8.61	-

3.2.4 Comparative performance evaluation of RCE-IFE with conventional FS methods

This section deals with the comparison of RCE-IFE with several widely used FS algorithms. The results were collected for five gene expression datasets: GDS2547, GDS3268, GDS3646, GDS3875, and GDS5037. We compared seven FS algorithms, which include 1) Minimum Redundancy Maximum Relevance (MRMR) [154], 2) Fast Correlation-Based Filter (FCBF) [155], 3) Information Gain (IG) [15], 4) Conditional Mutual Information Maximization (CMIM) [156], 5) SelectKBest (SKB), 6) eXtreme Gradient Boosting (XGBoost) [157], and 7) SVM-RFE [25]. In order to evaluate the quality of features obtained by FS methods, eight well-known classification algorithms were applied: 1) Adaboost, 2) Decision Tree (DT), 3) Logitboost, 4) RF, 5) Support Vector Classifier (SVC), 6) Stacking Classifier (base: Logitboost, k-Nearest Neighbour (KNN), final: RF), 7) Stacking Classifier (base: Logitboost, SVC, final: Logistic Regression), and 8) XGBClassifier.

The implementations were carried out through the skfeature and sklearn libraries in python [158]. The settings on hyper parameters of the classifiers can be found in Table A.5. The number of top genes selected by the FS algorithms was determined according to the average number of genes obtained by RCE-IFE for two clusters. In other words, the number of genes selected by all FS algorithms was kept the same for a fair comparison. Table 3.6 presents the results for the aforementioned FS algorithms and classifiers with the same number of selected genes. RCE-IFE is superior to the tested FS

methods in most cases in terms of AUC (refer to the Average column). While XGBoost with RF reaches the best average performance (79%), the average performance of RCE-IFE (76%) is either quite comparable or dominant over other methods. Out of 49 prediction performances, only 6 show the same or slightly better performance than RCE-IFE. MRMR, FCBF, IG, and CMIM obtain their highest performances with XGB but fall far short of RCE-IFE. However, SKB and XGBoost give competitive results with XGBClassifier. Overall, RCE-IFE outperforms FS algorithms substantially with different classifiers.

Table 3.6 Mean AUC values after comparing with popular FS algorithms for the same number of selected genes

FS Type	#selected genes	51	41	26	26	22	Average
	Classifier	GDS2547	GDS3268	GDS3646	GDS3875	GDS5037	
MRMR	Adaboost	0.49 ± 0.15	0.47 ± 0.16	0.52 ± 0.28	0.46 ± 0.32	0.45 ± 0.32	0.48 ± 0.25
	DT	0.49 ± 0.10	0.51 ± 0.12	0.50 ± 0.20	0.48 ± 0.21	0.50 ± 0.20	0.50 ± 0.17
	Logitboost	0.49 ± 0.15	0.48 ± 0.15	0.51 ± 0.25	0.47 ± 0.29	0.44 ± 0.32	0.48 ± 0.23
	RF	0.49 ± 0.15	0.50 ± 0.16	0.48 ± 0.29	0.44 ± 0.26	0.45 ± 0.26	0.47 ± 0.22
	SVC	0.50 ± 0.17	0.49 ± 0.15	0.59 ± 0.27	0.55 ± 0.32	0.49 ± 0.31	0.52 ± 0.24
	Stacking (Logitboost + KNN)	0.52 ± 0.15	0.49 ± 0.16	0.47 ± 0.28	0.52 ± 0.30	0.56 ± 0.31	0.51 ± 0.24
	Stacking (SVC + KNN)	0.51 ± 0.14	0.50 ± 0.16	0.50 ± 0.27	0.53 ± 0.30	0.54 ± 0.31	0.52 ± 0.24
	XGBClassifier	0.52 ± 0.14	0.75 ± 0.12	0.58 ± 0.26	0.46 ± 0.27	0.65 ± 0.29	0.59 ± 0.22
FCBF	Adaboost	0.61 ± 0.17	0.50 ± 0.14	0.48 ± 0.32	0.53 ± 0.28	0.43 ± 0.30	0.51 ± 0.24
	DT	0.57 ± 0.11	0.50 ± 0.12	0.47 ± 0.21	0.52 ± 0.21	0.48 ± 0.22	0.51 ± 0.17
	Logitboost	0.60 ± 0.16	0.49 ± 0.14	0.49 ± 0.33	0.50 ± 0.29	0.48 ± 0.31	0.51 ± 0.25
	RF	0.64 ± 0.15	0.49 ± 0.15	0.49 ± 0.30	0.51 ± 0.31	0.42 ± 0.31	0.51 ± 0.24
	SVC	0.57 ± 0.19	0.50 ± 0.16	0.52 ± 0.30	0.52 ± 0.29	0.57 ± 0.26	0.54 ± 0.24
	Stacking (Logitboost + KNN)	0.62 ± 0.16	0.49 ± 0.17	0.46 ± 0.30	0.48 ± 0.28	0.43 ± 0.31	0.50 ± 0.24
	Stacking (SVC + KNN)	0.67 ± 0.14	0.52 ± 0.14	0.50 ± 0.32	0.48 ± 0.30	0.43 ± 0.30	0.52 ± 0.24
	XGBClassifier	0.64 ± 0.18	0.62 ± 0.15	0.51 ± 0.24	0.34 ± 0.28	0.65 ± 0.31	0.55 ± 0.23

IG	Adaboost	0.77 ± 0.11	0.70 ± 0.14	0.53 ± 0.28	0.54 ± 0.28	0.73 ± 0.27	0.65 ± 0.22
	DT	0.66 ± 0.12	0.60 ± 0.14	0.50 ± 0.25	0.51 ± 0.21	0.68 ± 0.22	0.59 ± 0.19
	Logitboost	0.79 ± 0.12	0.72 ± 0.12	0.53 ± 0.28	0.50 ± 0.30	0.74 ± 0.25	0.66 ± 0.21
	RF	0.79 ± 0.11	0.71 ± 0.13	0.53 ± 0.26	0.55 ± 0.25	0.79 ± 0.23	0.67 ± 0.20
	SVC	0.59 ± 0.26	0.33 ± 0.14	0.49 ± 0.26	0.46 ± 0.30	0.31 ± 0.25	0.44 ± 0.24
	Stacking (Logitboost + KNN)	0.73 ± 0.12	0.57 ± 0.15	0.46 ± 0.26	0.53 ± 0.27	0.65 ± 0.29	0.59 ± 0.22
	Stacking (SVC + KNN)	0.76 ± 0.11	0.62 ± 0.19	0.47 ± 0.26	0.57 ± 0.29	0.61 ± 0.32	0.61 ± 0.23
	XGBClassifier	0.80 ± 0.11	0.74 ± 0.12	0.51 ± 0.27	0.51 ± 0.30	0.78 ± 0.24	0.67 ± 0.21
CMIM	Adaboost	0.51 ± 0.17	0.58 ± 0.15	0.51 ± 0.26	0.71 ± 0.26	0.83 ± 0.24	0.63 ± 0.22
	DT	0.53 ± 0.13	0.57 ± 0.12	0.52 ± 0.21	0.79 ± 0.18	0.73 ± 0.20	0.63 ± 0.17
	Logitboost	0.54 ± 0.17	0.58 ± 0.14	0.50 ± 0.27	0.79 ± 0.24	0.81 ± 0.24	0.64 ± 0.21
	RF	0.58 ± 0.17	0.56 ± 0.14	0.54 ± 0.27	0.84 ± 0.23	0.74 ± 0.27	0.65 ± 0.22
	SVC	0.46 ± 0.17	0.45 ± 0.14	0.50 ± 0.26	0.22 ± 0.27	0.45 ± 0.27	0.42 ± 0.22
	Stacking (Logitboost + KNN)	0.50 ± 0.17	0.47 ± 0.15	0.49 ± 0.28	0.64 ± 0.26	0.57 ± 0.32	0.53 ± 0.24
	Stacking (SVC + KNN)	0.56 ± 0.17	0.43 ± 0.13	0.51 ± 0.29	0.64 ± 0.29	0.59 ± 0.30	0.55 ± 0.24
	XGBClassifier	0.56 ± 0.15	0.64 ± 0.13	0.44 ± 0.27	0.80 ± 0.24	0.84 ± 0.21	0.66 ± 0.20
SKB	Adaboost	0.80 ± 0.11	0.79 ± 0.10	0.70 ± 0.25	0.73 ± 0.23	0.72 ± 0.26	0.75 ± 0.19
	DT	0.67 ± 0.12	0.63 ± 0.13	0.61 ± 0.22	0.57 ± 0.18	0.60 ± 0.18	0.62 ± 0.17
	Logitboost	0.79 ± 0.11	0.80 ± 0.10	0.75 ± 0.24	0.73 ± 0.24	0.79 ± 0.24	0.77 ± 0.19
	RF	0.81 ± 0.11	0.80 ± 0.10	0.74 ± 0.24	0.78 ± 0.20	0.78 ± 0.23	0.78 ± 0.18
	SVC	0.79 ± 0.12	0.23 ± 0.12	0.25 ± 0.25	0.29 ± 0.22	0.26 ± 0.24	0.36 ± 0.19
	Stacking (Logitboost + KNN)	0.72 ± 0.12	0.74 ± 0.13	0.74 ± 0.24	0.68 ± 0.23	0.76 ± 0.24	0.73 ± 0.19
	Stacking (SVC + KNN)	0.81 ± 0.10	0.80 ± 0.11	0.76 ± 0.25	0.66 ± 0.26	0.73 ± 0.25	0.75 ± 0.19
	XGBClassifier	0.81 ± 0.10	0.81 ± 0.12	0.70 ± 0.25	0.71 ± 0.22	0.76 ± 0.25	0.76 ± 0.19
XGBoost	Adaboost	0.79 ± 0.12	0.84 ± 0.10	0.64 ± 0.28	0.77 ± 0.20	0.67 ± 0.27	0.74 ± 0.19
	DT	0.64 ± 0.13	0.67 ± 0.11	0.55 ± 0.23	0.69 ± 0.20	0.62 ± 0.26	0.63 ± 0.20
	Logitboost	0.80 ± 0.11	0.86 ± 0.09	0.61 ± 0.28	0.80 ± 0.22	0.75 ± 0.26	0.76 ± 0.19
	RF	0.82 ± 0.11	0.85 ± 0.10	0.68 ± 0.26	0.82 ± 0.21	0.77 ± 0.25	0.79 ± 0.19
	SVC	0.78 ± 0.11	0.19 ± 0.11	0.38 ± 0.29	0.25 ± 0.23	0.27 ± 0.25	0.37 ± 0.20
	Stacking (Logitboost +	0.78 ± 0.11	0.83 ± 0.10	0.62 ± 0.26	0.77 ± 0.21	0.74 ± 0.23	0.75 ± 0.18

	KNN)						
	Stacking (SVC + KNN)	0.80 ± 0.11	0.80 ± 0.11	0.64 ± 0.27	0.75 ± 0.22	0.74 ± 0.29	0.75 ± 0.20
	XGBClassifier	0.81 ± 0.12	0.85 ± 0.10	0.63 ± 0.27	0.83 ± 0.20	0.76 ± 0.26	0.78 ± 0.19
SVM-RFE	SVM	0.78 ± 0.08	0.83 ± 0.07	0.74 ± 0.16	0.70 ± 0.17	0.72 ± 0.19	0.75 ± 0.13
RCE-IFE	RF	0.80 ± 0.11	0.76 ± 0.13	0.67 ± 0.25	0.83 ± 0.14	0.75 ± 0.22	0.76 ± 0.17

3.2.5 Comparative performance evaluation of RCE-IFE with Multi-stage algorithm

In this section, RCE-IFE is tested with four widely used cancer-related datasets that are readily available. We compared RCE-IFE with Multi-stage algorithm [114] due to its similarity to our approach. While both methods share some common steps, the way these steps are performed differs, as Multi-stage employs SVM-RFE and adopts a distinct strategy for cluster elimination. Table 3.7 presents the performance metrics obtained for RCE-IFE and Multi-stage on four datasets. The metrics for Multi-stage are extracted from the original study, where performance was reported as the average accuracy of the top 60 genes. For RCE-IFE, the numbers on the right of the “||” symbol refer to the average number of selected genes. As shown in Table 3.7, RCE-IFE achieves accuracy rates comparable to those of Multi-stage, with insignificant differences. Moreover, except for the Leukemia dataset, the number of genes selected by RCE-IFE is considerably below 60 in all cases. These findings imply that RCE-IFE has a very competitive performance in terms of accuracy measure and outperforms Multi-stage in terms of gene reduction.

Table 3.7 Comparative performance evaluation of RCE-IFE with SVM-RFE: Mean AUC and mean size of the feature subsets on metagenomics datasets

Datasets	Multi-stage	RCE-IFE
Breast	82.84	80.89 44
DLBCL	88.74	87.67 34
Leukemia	97.59	96.33 70
Prostate	93.45	92.18 38
Average	90.66	89.27 46.5

3.2.6 Comparative performance evaluation of RCE-IFE with other

biological domain knowledge-based G-S-M tools

We have comparatively evaluated RCE-IFE with three biological domain knowledge-based G-S-M tools, i.e., mirGediNET [129], 3Mint [108], and miRcorrNet [103], in terms of several performance metrics. These tools integrate biological knowledge into feature grouping and score the feature groups using a classifier. For comparison purposes, we employed the TCGA-BRCA dataset used in the aforementioned tools, available on the Genomic Data Commons (GDC) repository hosted by the National Cancer Institute (NCI). This dataset is a type of miRNA expression with the reads mapped to GRCh38 and downloaded from the UCSC Xena repository (<https://xenabrowser.net/datapages/>) [142]. For this dataset, tumor samples were filtered such that Luminal A and Luminal B subtypes (248 ER+/PR+/PR- samples) were considered positive (LumAB), while Her2-enriched and Basal-like subtypes (124 ER-/PR- samples) were considered negative (Her2Basal).

We selected the best performance metrics for each tool based on their highest AUC values for the TCGA-BRCA molecular subtype dataset and compared them with RCE-IFE. MirGediNET, 3Mint, and RCE-IFE obtain these AUC values via selecting a similar number of features, i.e., 9.6, 13.6, and 12.4 features, respectively, when averaged over 100 iterations. In contrast, miRcorrNet selects a larger number of features, i.e., 38.2 on average. Figure 3.5 plots several performance metrics obtained with these four tools when tested on the TCGA-BRCA molecular subtype dataset. As apparent in Figure 3.5, the performance measurements of RCE-IFE, miRcorrNet, 3Mint, and mirGediNET on the TCGA-BRCA molecular subtype dataset are close, implying that these tools are comparable. However, the outcome is different for each tool as each has its own benefit and is developed to uncover the significant groups based on specific biological knowledge.

3.2.7 Biological validation of the RCE-IFE findings

In this section, we analyze the biological relevancy and significance of the selected miRNAs for the TCGA-BRCA dataset introduced in the previous section. Breast cancer is the most commonly diagnosed malignancy worldwide and it is the leading cause of

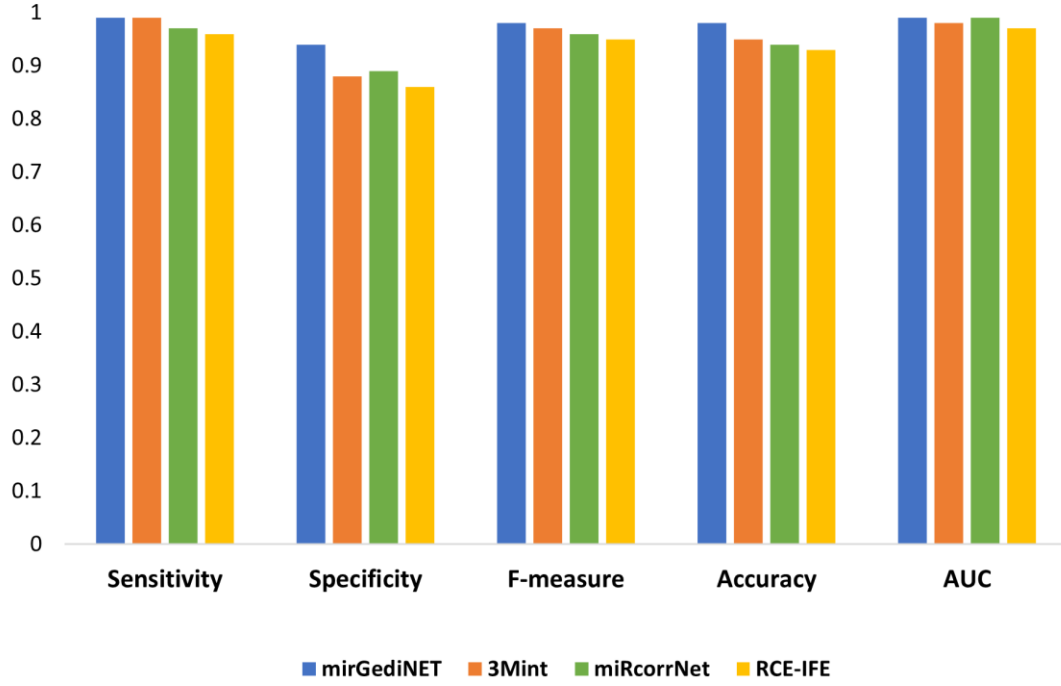


Figure 3.5 Comparative performance evaluation of RCE-IFE with three biological domain knowledge-based G-S-M tools

cancer-related mortality among women [159]. The top 5 miRNAs selected by our method on the TCGA-BRCA molecular subtype dataset are listed in Table 3.8. The top feature, miR-190b, was reported to be upregulated in patients with ER+ breast cancer, and its dysregulation plays a significant role in the initiation and progression of breast cancer [160]. This miRNA was found to suppress breast cancer metastasis by targeting SMAD2 [161]. MiR-135b is a key regulator in breast cancer and promotes tumor growth, invasion, and metastasis [162]. It serves as an oncogene, and due to its effect in modulating critical signaling pathways, it is considered a potential biomarker in breast cancer and a promising target for therapeutic intervention [163]. The expression of miR-18a correlates with ER-breast tumors characterized by a high level of inflammation [164]. High expression of it is closely linked to basal-like breast cancer [165]. Downregulation of miR-505 promotes cellular processes such as cell proliferation, migration, and invasion. Moreover, the low expression level of this miRNA is associated with poor prognosis in breast cancer patients [166]. MiR-934 expression has a strong association with the overall survival of breast cancer patients and promotes cell metastasis by targeting PTEN [167]. In addition, mir934 inhibition suppresses the migration capability of tumor cells to a certain extent in patients with triple-negative breast cancer [168].

3.2.8 Consistency of the selected features across different runs

To evaluate the consistency of miRNAs selected by the proposed algorithm, we run RCE-IFE on the TCGA-BRCA molecular subtype dataset three times, where each run consists of 100 iterations. Top selected miRNAs were compared among successive runs. Figure 6 depicts the overlaps of the top (A) 5 miRNAs, (B) 10 miRNAs, and (C) 20 miRNAs between three different runs. As apparent in Figure 6, the top 5 miRNAs are the same miRNAs for all the runs. Among the top 10 miRNAs selected in three different runs, nine are commonly identified in all three runs, and one miRNA is common among the two runs. Only one miRNA is non-overlapping with any other runs. We encounter a similar situation when the top 20 features are selected in three different runs. 19 miRNAs are commonly detected in all runs, one miRNA is shared between the results of the two runs, and one miRNA is unique to one run. These results reveal the large-scale consistency of the miRNAs selected by the RCE-IFE algorithm across different runs.

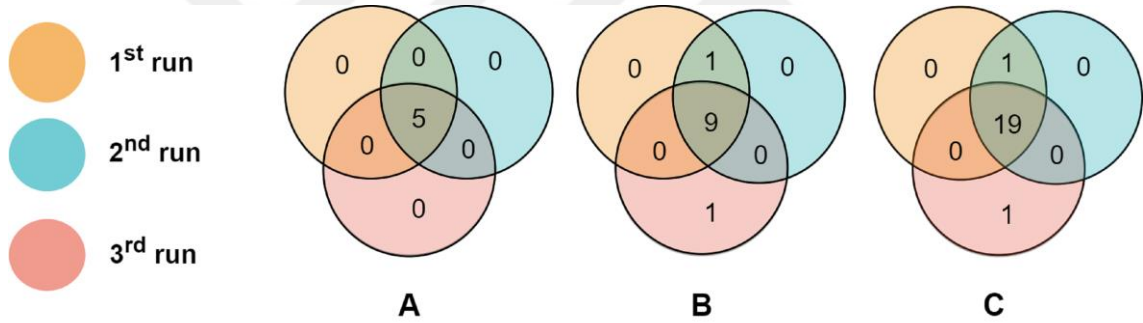


Figure 3.6 Overlaps between the top (A) 5, (B) 10, (C) 20 miRNAs selected in different runs

3.3 Discussion

In this study, we address the challenge of the FS task using a feature grouping-based strategy. We propose RCE-IFE, which involves cluster elimination followed by removing features from surviving clusters to retain non-redundant and strongly relevant features. This approach leads to further dimensionality reduction, improved feature subset quality, enhanced model performance, and reduced computation time. We have conducted different experiments on publicly available datasets from several biological domains to assess the performance of RCE-IFE.

RCE-IFE, RCE-IFE-SVM (SVM adapted version of RCE-IFE), and SVM-RCE yield comparable average AUCs of 0.85, 0.84, and 0.83, respectively, when averaged

over different gene expression, miRNA expression, and methylation datasets; however, RCE-IFE achieves this performance via selecting the fewest features and the least execution time. When tested on seven different metagenomics datasets, RCE-IFE and SVM-RCE obtain average AUCs of 0.79 and 0.68, respectively. In these experiments on metagenomics dataset, RCE-IFE demonstrates a significant reduction in the size of feature subsets. In addition, RCE-IFE outperforms various popular FS methods, including MRMR, FCBF, IG, CMIM, SKB, and XGBoost, achieving an average AUC of 0.76 on five gene expression datasets. Compared to a similar tool, Multi-stage, RCE-IFE acquires a similar average accuracy rate of 89.27% using a lower number of features on four cancer-related datasets.

Additionally, we show that the performance of RCE-IFE is comparable with other biological domain knowledge-based G-S-M tools (mirGediNET, 3Mint, and miRcorrNet) on the TCGA-BRCA dataset. We verify in scientific literature that all of the identified top 5 miRNAs (miR-190b, miR-135b, miR-18a, miR-505, and miR-934) play significant roles in disease progression and prognosis, contributing to the elucidation of the molecular mechanisms of breast cancer and the development of treatment strategies. Finally, we indicate that the proposed algorithm is capable of selecting features with a high degree of consistency across multiple runs. Our findings suggest that the proposed method is powerful in that it provides robust model prediction, achieves a substantial reduction in feature dimension, and ensures both feature relevancy and consistency.

Chapter 4

Determining representative features via grouping, scoring and modeling (G-S-M)

In this study, we propose a G-S-M based method which involves steps of grouping features, scoring feature clusters, and creating the classification model. In our approach, we utilize DisGeNET [98] as prior knowledge to group features into disease groups. After grouping, we score these disease groups using RF algorithm. Next, we select top features from each group as group representatives. Eventually, we cumulatively select top groups and measure different performance metrics. That is; top group is firstly selected, then top two groups are selected and this continues until top 10 groups are selected.

4.1 Datasets

The datasets used in our G-S-M based project are accessible through GEO [140] database. All datasets correspond to several types of complex diseases such as glioma, colitis and cancer. Table 4.1 summarizes 10 datasets from the domain of gene expression.

Table 4.1 Main characteristics of 10 gene expression datasets

Dataset	#samples	Label: count	#features
GDS1962	178	pos:157, neg:23	12600
GDS2545	171	pos:90, neg:81	12581
GDS2547	164	pos:89, neg:75	12647
GDS2771	192	pos:102, neg:90	22216
GDS3257	107	pos:58, neg:49	22226
GDS3268_v2	200	pos:129, neg:71	40992
GDS3837	120	pos:60, neg:60	30623
GDS4206	197	pos:40, neg:157	54624
GDS4516_4718	148	pos:104, neg:44	54676
GDS5499	140	pos:99, neg:41	48804

4.2 The proposed G-S-M based approach

As stated previously, we benefit from external knowledge (DisGeNET) to group genes into disease groups. DisGeNET is a publicly available repository that associates genes with human diseases. Figure 4.1 illustrates the scenario of the grouping stage. After creating disease groups, these groups are assigned scores by RF and the model is learnt by the top disease groups. The performance results are collected cumulatively starting from the top scoring group. The workflow of the G-S-M based project is depicted in Figure 4.2.

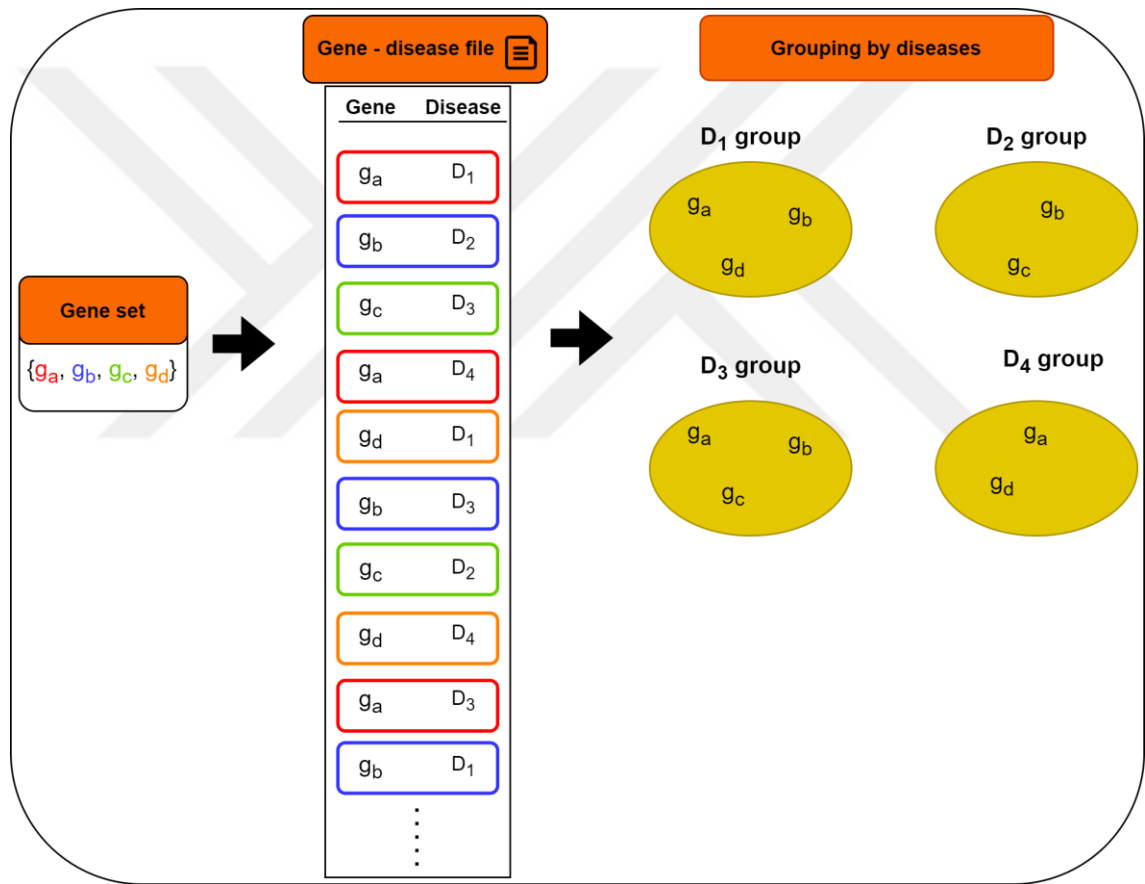


Figure 4.1 Creating feature groups using prior knowledge

4.3 Results

This section presents performance of our G-S-M based method compared to standard G-S-M approach. The difference between two methods is that standard G-S-M approach is lack of representative feature selection and considers all genes in a cluster. In our G-S-M method, we maximally selected top 5 genes (if exist) from each disease group.

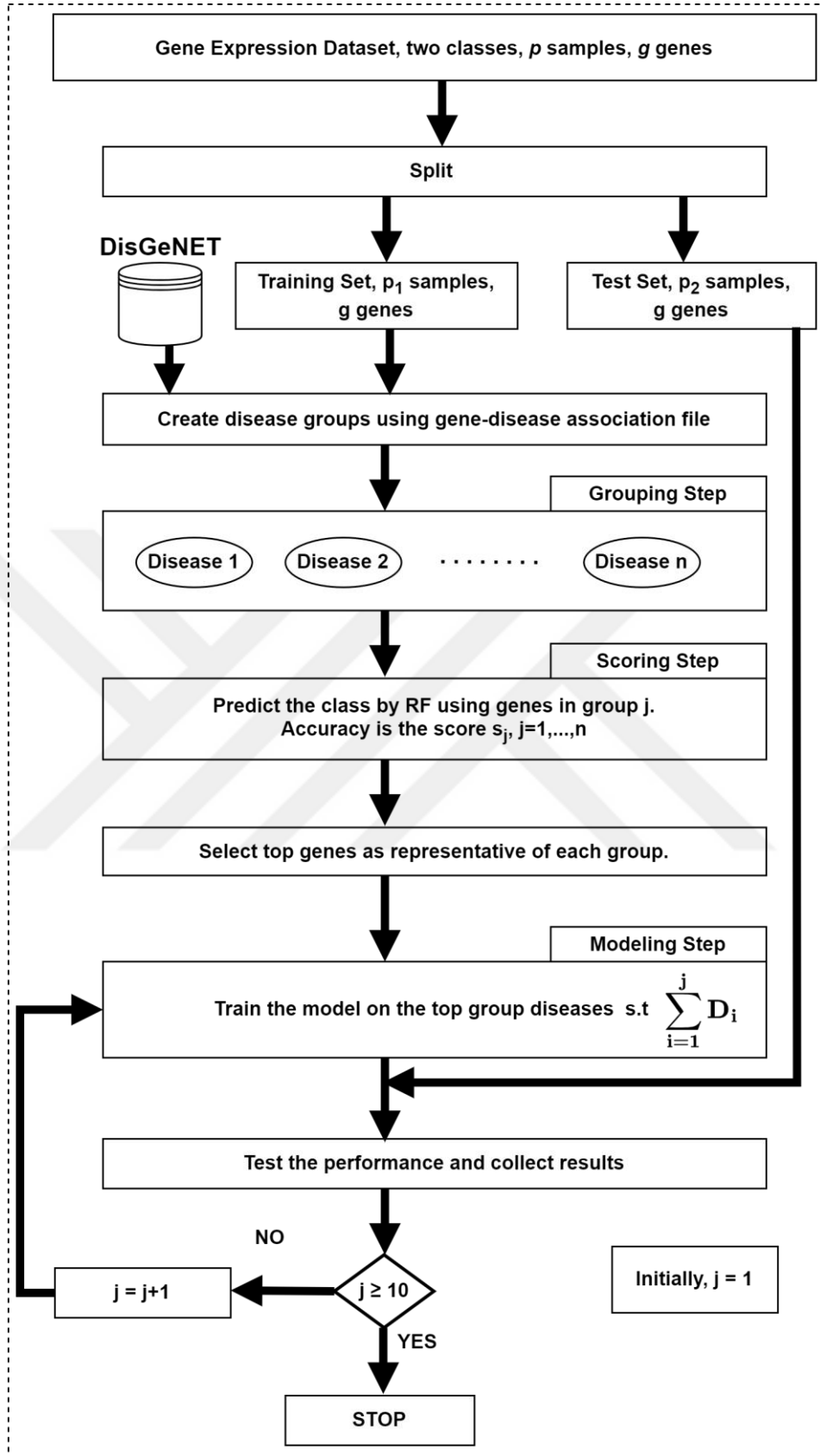


Figure 4.2 The framework of the G-S-M_Rep approach

Results were collected for 10 iterations. We denoted our G-S-M approach as G-S-M_Rep

to imply that we select representative genes from disease groups.

4.3.1 Comparison of G-S-M_Rep on gene expression datasets

Table 4.2 shows the results for 10 gene expression datasets in terms of AUC and feature size. While G-S-M_Rep increases the average AUC performance by 1%, it substantially reduces the feature size. Indeed, the amount of decrease is quite large and this shows that our algorithm is capable of selecting important genes for classification.

Table 4.2 Comparison between standard G-S-M and G-S-M_Rep for top two groups

Dataset	AUC		#Features	
	Standard G-S-M	G-S-M_Rep	Standard G-S-M	G-S-M_Rep
GDS1962	0.97	0.99	44.3	8.2
GDS2545	0.77	0.85	98.5	8.7
GDS2547	0.81	0.81	80.9	8.8
GDS2771	0.68	0.68	100.6	8.4
GDS3257	1	0.99	80.7	8.9
GDS3268_v2	0.77	0.70	88.9	7.7
GDS3837	0.97	0.95	100.6	9.4
GDS4206	0.57	0.62	49.8	6
GDS4516_4718	1	1	28.1	9
GDS5499	0.95	0.96	60.1	8.6
Average	0.85	0.86	73.25	8.37

We also determined the number of common genes found by the two methods. Table 4.3 presents the total number of genes and the number of common genes for top five groups. As apparent in Table 4.3, most genes found by G-S-M_Rep overlap with genes found by the standard G-S-M. These results also indicate that G-S-M_Rep is capable of removing irrelevant or redundant features while retaining informative features.

Table 4.3 Gene counts for standard G-S-M and G-S-M_Rep methods for top five groups

Dataset	Standard G-S-M	G-S-M_Rep	# of Common Genes
GDS1962	576	61	49
GDS2545	1324	38	26
GDS2547	455	48	40
GDS2771	985	69	59
GDS3257	557	56	38

GDS3268_v2	1047	88	65
GDS3837	417	46	41
GDS4206	831	100	67
GDS4516_4718	384	27	24

4.3.2 Pathway analysis

In this section, we collected genes from top five groups for each dataset and performed functional enrichment analysis using STRING [169]. Then, we aggregated all the detected pathways for all the datasets and selected top 10 KEGG pathways along with their associated genes. Figure 4.3 shows the pathway – gene network built using Cytoscape [170]. Although datasets belong to different disease types, it is seen that G-S-M_Rep can detect associations among them.

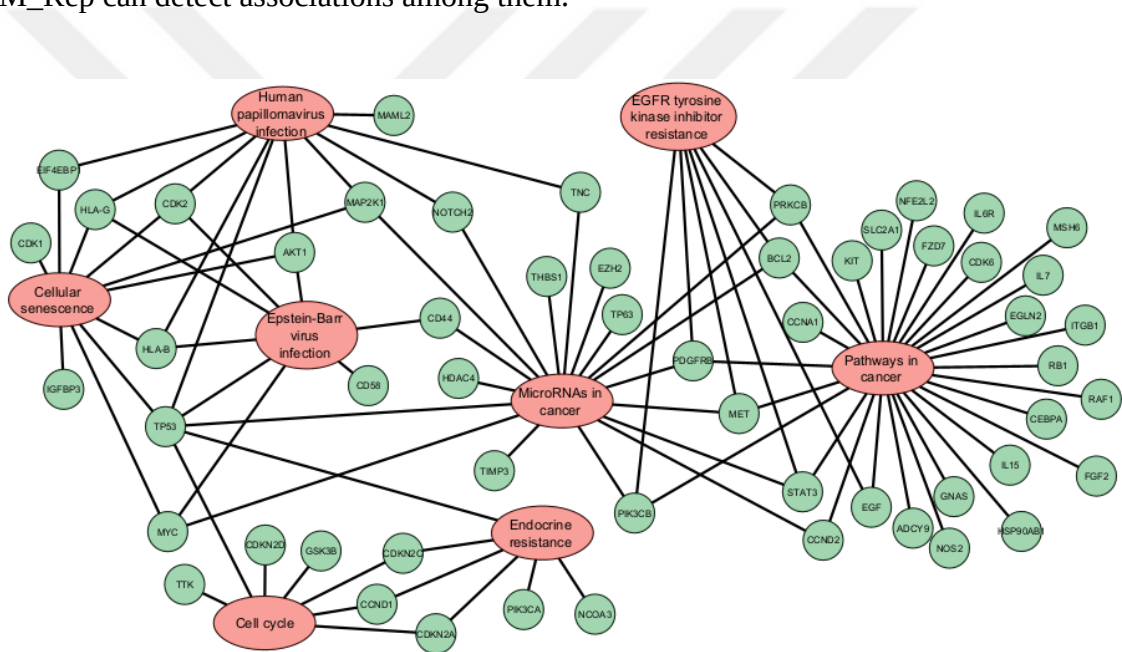


Figure 4. 3 Pathway – gene network of all datasets

4.3.3 Hub gene analysis

In this part, we wanted to learn if genes detected by G-S-M_Rep are mainly hub genes of Standard G-S-M genes. We considered genes from top five diseases for each dataset. To detect hub genes, we constructed a network of genes from Standard G-S-M genes using STRING. Next, we used the Cytohubba plugin of Cytoscape to find hub genes in this network. Finally, top 50 hub genes in this network are intersected with G-S-M_Rep genes. Table 4. 4 presents the number of detected hub genes by G-S-M_Rep for each dataset. Even though the number of detected hub genes is not high, G-S-M_Rep

finds a certain amount in all datasets. Note also that some hub genes in the network may have very close degree of connectivity and that some undetected genes can be the first neighbor of hub genes.

Table 4. 4 Number of hub genes in the network of Standard G-S-M genes

Dataset	#hub genes in top 50 rankings
GDS1962	4
GDS2545	2
GDS2547	7
GDS2771	8
GDS3257	6
GDS3268	13
GDS3837	7
GDS4206	11
GDS4516_4718	4
GDS5499	5

4.3.4 Gene analysis for GDS2545 dataset

We wanted to examine the gene behavior across the iterations to find the important genes for GDS2545 prostate cancer dataset. We considered genes from top five groups for this dataset. Figure 4.4 visualizes the heatmap for this analysis. In the heatmap, dark regions show high rank and yellow regions imply absence of the particular gene in the particular iteration. As seen from the heatmap, some genes such as MEN1, EZH2 survive through all iterations and generally have high rank. This tells us that these genes seem to be significant for this dataset. Indeed, there is clear evidence in the literature that these genes are associated with prostate cancer [171-172].

4.4 Discussion

This study is still ongoing and we continue to deepen our analysis and collect results. Our initial results are quite promising as presented above. In later analyses, we will consider selecting different numbers of features from disease groups and seek to make a conclusion. Moreover, we plan to exploit other types of grouping strategies, such as microRNA and its target genes or miRNA – disease associations to deepen our analyses. We will surely focus on other issues, e.g., studying on disease names, detecting active subnetworks etc. to conduct more biological analyses.

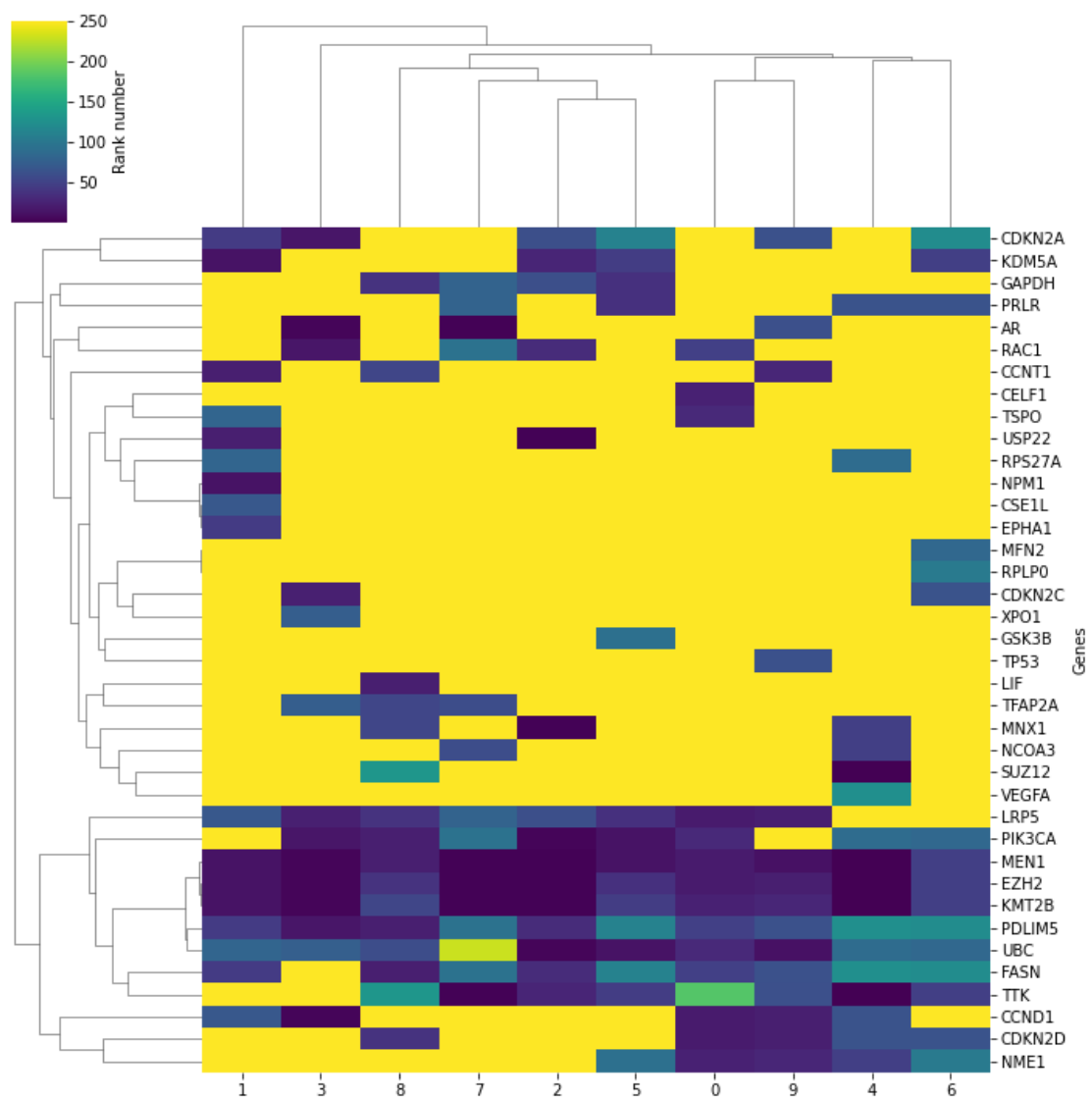


Figure 4. 4 Heatmap of genes through 10 iterations

Chapter 5

Conclusions and Future Prospects

5.1 Conclusions

The advances in high-throughput technologies have generated large high-dimensional data sets in many applications. The inevitable presence of redundant and noisy features increases computational complexity and degrades classifier capability. Hence, FS has become a required pre-processing step in itself as a primary concern for a long time. We firstly examined the effect of different clustering algorithms on SVM-RCE and found that effect of K-means gives the best effect among them. Then, we presented the first developed approach RCE-IFE in this thesis. RCE-IFE performs FS by eliminating both clusters and features within clusters. Initially, samples are partitioned for training and test and then all features are applied t-test to filter top 1000 features. K-means clustering is applied to create feature clusters. These clusters are assigned scores according to their classification capability by RF and a certain number of clusters with low scores are removed. The features from the remaining clusters are assembled and learnt by the model. Finally, the model performance is tested using the initial reserved test set. Feature clustering and the subsequent steps are performed recursively and performance evaluation is obtained for different cluster numbers in this way. In our analyses with RCE-IFE, we especially observed high power of RCE-IFE in terms of feature reduction without deteriorating classifier performance. As a first attempt, we compared RCE-IFE, RCE-IFE-SVM (RCE-IFE with SVM adaptation) and SVM-RCE. Experiments show that although RCE-IFE and RCE-IFE-SVM exhibit similar classification performance to the crude method, SVM-RCE, they accomplish it with a higher dimensionality reduction, i.e., with a significantly smaller number of features. Hence, IFE proves to be an effective step in dimensionality reduction without compromising classifier performance. Furthermore, RCE-IFE approaches are faster than SVM-RCE, another contribution of IFE. Comparing RCE-IFE to SVM-RCE on

metagenomics datasets indicate that RCE-IFE is more robust and more effective in both predictive performance and dimension reduction. Overall, the results show that RCE-IFE outperforms SVM-RCE in terms of classification, number of retained features and running time. We also compared RCE-IFE with various traditional FS methods employing different classification algorithms. The results indicate the dominance of our proposed approach over several FS methods. Among them; MRMR, FCBF, IG and CMIM give lower performance than RCE-IFE for all classifiers. Only SKB and XGBoost provide the same or slightly better performances with some classifiers. In addition, we compared RCE-IFE with Multi-stage, an approach involving similar steps to our algorithm. Once again, RCE-IFE yielded very good performance results and showed to be competitive. The accuracies provided by both approaches are very close to each other, and RCE-IFE achieves it with significantly lower number of features compared to Multi-stage in diverse datasets. Additionally, we have comparatively evaluated RCE-IFE with other biological domain knowledge-based Grouping-Scoring-Modeling (G-S-M) tools (mirGediNET, 3Mint, and miRcorrNet) and conducted biological validation of the selected features. We further show the high consistency of the features selected by RCE-IFE across multiple runs. Overall, via testing through diverse biological datasets concerning various diseases, the experimental findings show the effectiveness of the proposed RCE-IFE method with respect to predictive power, reduced feature subset selection, feature relevancy, and consistency of selected features across different runs.

Regarding the second proposed approach in this thesis, we designed G-S-M_Rep method which utilizes domain knowledge, i.e., incorporates prior knowledge in FS. DisGeNET is used as domain knowledge to create disease groups. These disease groups are given scores for the class prediction by RF. A fixed number of features from each disease group is selected and the model is trained on the top group diseases cumulatively. For instance, the model is trained on the top group 1 first, then on the top groups 1 and 2, and so on... For each cumulation, the model performance is recorded using various metrics. The algorithm stops after the model is tested by top 10 disease groups. This study is still ongoing and our initial results make us hopeful for the rest in that G-S-M_Rep can detect significant biomarkers and provides satisfactory results with few features.

To conclude, we aim to obtain good performance results with few features in favor of biomarker detection of a specific disease. The proposed approaches presented here may play a significant role in elucidating the mechanisms underlying a disease and contributing to drug therapy.

5.2 Societal Impact and Contribution to Global Sustainability

ML in bioinformatics is transforming the way biological data is analyzed, contributing significantly to societal advancement and sustainability. One of the most prominent impacts lies in precision medicine. ML models enable the identification of biomarkers crucial for early disease diagnosis and personalized treatment strategies. By tailoring therapies to individual patients, these innovations improve health outcomes, reduce adverse effects, and lower healthcare costs.

Additionally, ML enhances public health by analyzing large datasets to predict disease outbreaks, identify at-risk populations, and optimize intervention strategies. These tools empower public health officials to allocate resources effectively and respond proactively to global health challenges. For example, during pandemics, machine learning has played a critical role in analyzing viral mutations, forecasting infection rates, and developing vaccines.

ML also democratizes healthcare by making advanced diagnostic tools and predictive models more accessible, particularly in underserved or resource-limited areas. This reduces health disparities and ensures that cutting-edge scientific advancements benefit a broader spectrum of society.

From a sustainability perspective, machine learning reduces the dependency on resource-intensive experimental methods by extracting meaningful insights from existing datasets. This optimization minimizes energy consumption, material waste, and costs associated with large-scale biological experiments. Furthermore, it accelerates drug discovery processes, reducing the time and resources required to develop effective treatments for global diseases.

Beyond healthcare, bioinformatics applications in agriculture and environmental science also benefit from machine learning. These models assist in understanding genetic diversity, improving crop yields, and conserving biodiversity, which are critical for sustainable agricultural practices and environmental preservation.

Overall, the integration of machine learning in bioinformatics not only advances scientific research and healthcare but also supports several Sustainable Development Goals (SDGs), including good health and well-being (SDG 3), reduced inequalities (SDG 10), and climate action (SDG 13). Its wide-ranging applications underline its importance as a tool for fostering societal progress and addressing global sustainability challenges.

5.3 Future Prospects

As for future work concerning RCE-IFE, we intend to experiment with additional types of datasets from diverse domains, such as text or image. We also plan to implement the selection of a fixed number of representative features from each cluster to further reduce the computational overhead. Lastly, the proposed framework is intended to be adaptable for multiclass classification and multi-label FS problems.

In G-S-M_Rep study, we intend to explore the selection of varying numbers of features from disease groups to derive meaningful conclusions. Additionally, we aim to investigate alternative grouping strategies, such as microRNA and its target genes or microRNA-disease associations, to enhance the depth of our analyses. Furthermore, our focus will extend to other aspects, including analyzing disease names, identifying active subnetworks, and conducting comprehensive biological investigations.

BIBLIOGRAPHY

- [1] H. M. Abdulwahab, S. Ajitha, and M. A. N. Saif, “Feature selection techniques in the context of big data: taxonomy and analysis,” *Appl. Intell.*, vol. 52, no. 12, pp. 13568–13613, Sep. 2022, doi: 10.1007/s10489-021-03118-3.
- [2] B. Venkatesh and J. Anuradha, “A Review of Feature Selection and Its Methods,” *Cybern. Inf. Technol.*, vol. 19, no. 1, pp. 3–26, Mar. 2019, doi: 10.2478/cait-2019-0001.
- [3] A. Jovic, K. Brkic, and N. Bogunovic, “A review of feature selection methods with applications,” in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia: IEEE, May 2015, pp. 1200–1205. doi: 10.1109/MIPRO.2015.7160458.
- [4] C. Kuzudisli, B. Bakir-Gungor, N. Bulut, B. Qaqish, and M. Yousef, “Review of feature selection approaches based on grouping of features,” *PeerJ*, vol. 11, p. e15666, Jul. 2023, doi: 10.7717/peerj.15666.
- [5] M. M. Hassan, S. Mollick, and F. Yasmin, “An unsupervised cluster-based feature grouping model for early diabetes detection,” *Healthc. Anal.*, vol. 2, p. 100112, 2022.
- [6] G. H. John, R. Kohavi, and K. Pfleger, “Irrelevant Features and the Subset Selection Problem,” in *Machine Learning Proceedings 1994*, Elsevier, 1994, pp. 121–129. doi: 10.1016/B978-1-55860-335-6.50023-4.
- [7] J. Wang, X. Wu, and C. Zhang, “Support vector machines based on K-means clustering for real-time business intelligence systems,” *Int. J. Bus. Intell. Data Min.*, vol. 1, no. 1, pp. 54–64, Jan. 2005, doi: 10.1504/IJBIDM.2005.007318.
- [8] L. Maokuan, C. Yusheng, and Z. Honghai, “Unlabeled data classification via support vector machines and k-means clustering,” in *Proceedings. International Conference on Computer Graphics, Imaging and Visualization, 2004. CGIV 2004.*, Jul. 2004, pp. 183–186. doi: 10.1109/CGIV.2004.1323982.
- [9] X. Shen and H.-C. Huang, “Grouping Pursuit Through a Regularization Solution Surface,” *J. Am. Stat. Assoc.*, vol. 105, no. 490, pp. 727–739, Jun. 2010, doi: 10.1198/jasa.2010.tm09380.
- [10] Y. Dai *et al.*, “Feature Grouping for No-reference Image Quality Assessment,” in *2022 7th International Conference on Automation, Control and Robotics Engineering (CACRE)*, Xi’an, China: IEEE, Jul. 2022, pp. 204–208. doi: 10.1109/CACRE54574.2022.9834184.
- [11] M. Sood, P. Angra, S. Verma, N. Jhanjhi, and others, “Efficient feature grouping for IDS using clustering algorithms in detecting known/unknown attacks,” in *Information security handbook*, CRC Press, 2022, pp. 103–116.
- [12] A. N. M. B. Rashid, M. Ahmed, L. F. Sikos, and P. Haskell-Dowland, “Cooperative co-evolution for feature selection in Big Data with random feature

- grouping,” *J. Big Data*, vol. 7, no. 1, p. 107, Dec. 2020, doi: 10.1186/s40537-020-00381-y.
- [13] L. AbdAllah, W. Khalifa, L. C. Showe, and M. Yousef, “Selection of Significant Clusters of Genes based on Ensemble Clustering and Recursive Cluster Elimination (RCE),” *J. Proteomics Bioinform.*, vol. 10, no. 8, 2017, doi: 10.4172/jpb.1000439.
 - [14] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” *Artif. Intell.*, vol. 97, no. 1–2, pp. 273–324, Dec. 1997, doi: 10.1016/S0004-3702(97)00043-X.
 - [15] M. A. Hall and L. A. Smith, “Practical feature subset selection for machine learning,” Conference held at Perth: Springer, Feb. 1998, pp. 181–191. [Online]. Available: <https://hdl.handle.net/10289/1512>
 - [16] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd edition. Cambridge, UK ; New York: Cambridge University Press, 2007.
 - [17] D. Nettleton, *Commercial Data Mining: Processing, Analysis and Modeling for Predictive Analytics Projects*, 1st edition. Amsterdam: Morgan Kaufmann, 2014.
 - [18] H. Liu and R. Setiono, “Chi2: Feature selection and discretization of numeric attributes,” in *Proceedings of 7th IEEE international conference on tools with artificial intelligence*, Ieee, 1995, pp. 388–391.
 - [19] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 1st ed. Wiley, 2005. doi: 10.1002/047174882X.
 - [20] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd edition. Burlington, MA: Morgan Kaufmann, 2011.
 - [21] Q. Song, J. Ni, and G. Wang, “A Fast Clustering-Based Feature Subset Selection Algorithm for High-Dimensional Data,” *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 1–14, Jan. 2013, doi: 10.1109/TKDE.2011.181.
 - [22] S. Visalakshi and V. Radha, “A literature review of feature selection techniques and applications: Review of feature selection in data mining,” in *2014 IEEE International Conference on Computational Intelligence and Computing Research*, Coimbatore, India: IEEE, Dec. 2014, pp. 1–6. doi: 10.1109/ICCIC.2014.7238499.
 - [23] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*. Boston, MA: Springer US, 1998. doi: 10.1007/978-1-4615-5689-3.
 - [24] H. Liu and L. Yu, “Toward integrating feature selection algorithms for classification and clustering,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, 2005.
 - [25] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene Selection for Cancer Classification using Support Vector Machines,” *Mach. Learn.*, vol. 46, no. 1/3, pp. 389–422, 2002, doi: 10.1023/A:1012487302797.
 - [26] J. Li *et al.*, “Feature Selection: A Data Perspective,” *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–45, Nov. 2018, doi: 10.1145/3136625.

- [27] R. Tibshirani, "Regression Shrinkage and Selection Via the Lasso," *J. R. Stat. Soc. Ser. B Methodol.*, vol. 58, no. 1, pp. 267–288, Jan. 1996, doi: 10.1111/j.2517-6161.1996.tb02080.x.
- [28] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: A review," *Data Classif. Algorithms Appl.*, p. 37, 2014.
- [29] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification And Regression Trees*, 1st ed. Routledge, 2017. doi: 10.1201/9781315139470.
- [30] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, no. 1–4, pp. 131–156, 1997, doi: 10.1016/S1088-467X(97)00008-5.
- [31] W. Pan, "A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments," *Bioinformatics*, vol. 18, no. 4, pp. 546–554, Apr. 2002, doi: 10.1093/bioinformatics/18.4.546.
- [32] C. Lazar *et al.*, "A Survey on Filter Techniques for Feature Selection in Gene Expression Microarray Analysis," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 9, no. 4, pp. 1106–1119, Jul. 2012, doi: 10.1109/TCBB.2012.33.
- [33] L. Talavera, "An Evaluation of Filter and Wrapper Methods for Feature Selection in Categorical Clustering," in *Advances in Intelligent Data Analysis VI*, vol. 3646, A. F. Famili, J. N. Kok, J. M. Peña, A. Siebes, and A. Feelders, Eds., in Lecture Notes in Computer Science, vol. 3646. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 440–451. doi: 10.1007/11552253_40.
- [34] N. El Aboudi and L. Benhlila, "Review on wrapper feature selection approaches," in *2016 International Conference on Engineering & MIS (ICEMIS)*, Agadir, Morocco: IEEE, Sep. 2016, pp. 1–5. doi: 10.1109/ICEMIS.2016.7745366.
- [35] S. Ma and J. Huang, "Penalized feature selection and classification in bioinformatics," *Brief. Bioinform.*, vol. 9, no. 5, pp. 392–403, Apr. 2008, doi: 10.1093/bib/bbn027.
- [36] D. Asir, S. Appavu, and E. Jebamalar, "Literature Review on Feature Selection Methods for High-Dimensional Data," *Int. J. Comput. Appl.*, vol. 136, no. 1, pp. 9–17, Feb. 2016, doi: 10.5120/ijca2016908317.
- [37] V. Bolón-Canedo and A. Alonso-Betanzos, "Ensembles for feature selection: A review and future trends," *Inf. Fusion*, vol. 52, pp. 1–12, Dec. 2019, doi: 10.1016/j.inffus.2018.11.008.
- [38] J. C. Ang, A. Mirzal, H. Haron, and H. N. A. Hamed, "Supervised, Unsupervised, and Semi-Supervised Feature Selection: A Review on Gene Selection," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 13, no. 5, pp. 971–989, Sep. 2016, doi: 10.1109/TCBB.2015.2478454.
- [39] U. M. Khaire and R. Dhanalakshmi, "Stability of feature selection algorithm: A review," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 4, pp. 1060–1073, Apr. 2022, doi: 10.1016/j.jksuci.2019.06.012.
- [40] F. Kamalov, F. Thabtah, and H. H. Leung, "Feature Selection in Imbalanced Data," *Ann. Data Sci.*, Jan. 2022, doi: 10.1007/s40745-021-00366-5.

- [41] S. Chormunge and S. Jena, "Correlation based feature selection with clustering for high dimensional data," *J. Electr. Syst. Inf. Technol.*, vol. 5, no. 3, pp. 542–549, Dec. 2018, doi: 10.1016/j.jesit.2017.06.004.
- [42] H. Liu, X. Wu, and S. Zhang, "Feature selection using hierarchical feature clustering," in *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*, Glasgow, Scotland, UK: ACM Press, 2011, p. 979. doi: 10.1145/2063576.2063716.
- [43] C. H. Park, "A Feature Selection Method Using Hierarchical Clustering," in *Mining Intelligence and Knowledge Exploration*, vol. 8284, R. Prasath and T. Kathirvalavakumar, Eds., in *Lecture Notes in Computer Science*, vol. 8284, Cham: Springer International Publishing, 2013, pp. 1–6. doi: 10.1007/978-3-319-03844-5_1.
- [44] D. Harris and A. Van Niekerk, "Feature clustering and ranking for selecting stable features from high dimensional remotely sensed data," *Int. J. Remote Sens.*, vol. 39, no. 23, pp. 8934–8949, Dec. 2018, doi: 10.1080/01431161.2018.1500730.
- [45] S. Yang, L. Yuan, Y.-C. Lai, X. Shen, P. Wonka, and J. Ye, "Feature grouping and selection over an undirected graph," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*, Beijing, China: ACM Press, 2012, p. 922. doi: 10.1145/2339530.2339675.
- [46] J. Martínez Sotoca and F. Pla, "Supervised feature selection by clustering using conditional mutual information-based distances," *Pattern Recognit.*, vol. 43, no. 6, pp. 2068–2081, Jun. 2010, doi: 10.1016/j.patcog.2009.12.013.
- [47] L. Yu, C. Ding, and S. Loscalzo, "Stable feature selection via dense feature groups," in *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 08*, Las Vegas, Nevada, USA: ACM Press, 2008, p. 803. doi: 10.1145/1401890.1401986.
- [48] R. A. Shah, Y. Qian, and G. Mahdi, "Group Feature Selection via Structural Sparse Logistic Regression for IDS," in *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Sydney, Australia: IEEE, Dec. 2016, pp. 594–600. doi: 10.1109/HPCC-SmartCity-DSS.2016.0089.
- [49] S. Petry, C. Flexeder, and G. Tutz, "Pairwise Fused Lasso," 2011, doi: 10.5282/UBM/EPUB.12164.
- [50] B. Sahu, S. Dehuri, and A. K. Jagadev, "Feature selection model based on clustering and ranking in pipeline for microarray data," *Inform. Med. Unlocked*, vol. 9, pp. 107–122, 2017, doi: 10.1016/j.imu.2017.07.004.
- [51] Z. Shang and M. Li, "Feature Selection Based on Grouped Sorting," in *2016 9th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou: IEEE, Dec. 2016, pp. 451–454. doi: 10.1109/ISCID.2016.1111.
- [52] K. Zhu and J. Yang, "A cluster-based sequential feature selection algorithm," in *2013 Ninth International Conference on Natural Computation (ICNC)*, Shenyang, China: IEEE, Jul. 2013, pp. 848–852. doi: 10.1109/ICNC.2013.6818094.

- [53] M. Alimoussa, A. Porebski, N. Vandenbroucke, R. Thami, and S. El Fkihi, "Clustering-based Sequential Feature Selection Approach for High Dimensional Data Classification:," in *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Online Streaming, --- Select a Country ---: SCITEPRESS - Science and Technology Publications, 2021, pp. 122–132. doi: 10.5220/0010259501220132.
- [54] M. Alimoussa, A. Porebski, N. Vandenbroucke, S. El Fkihi, and R. Oulad Haj Thami, "Compact Hybrid Multi-Color Space Descriptor Using Clustering-Based Feature Selection for Texture Classification," *J. Imaging*, vol. 8, no. 8, p. 217, Aug. 2022, doi: 10.3390/jimaging8080217.
- [55] W.-H. Au, K. C. Chan, A. K. Wong, and Y. Wang, "Attribute clustering for grouping, selection, and classification of gene expression data," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 2, no. 2, pp. 83–101, 2005.
- [56] E. Chitsaz, M. Taheri, and S. D. Katebi, "A fuzzy approach to clustering and selecting features for classification of gene expression data," in *Proceedings of the World Congress on Engineering*, 2008.
- [57] E. Chitsaz, M. Taheri, S. D. Katebi, and M. Z. Jahromi, "An improved fuzzy feature clustering and selection based on chi-squared-test," in *Proceedings of the international multiconference of engineers and computer scientists*, 2009, pp. 18–20.
- [58] Q. Liu, J. Zhang, J. Xiao, H. Zhu, and Q. Zhao, "A Supervised Feature Selection Algorithm through Minimum Spanning Tree Clustering," in *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, Limassol, Cyprus: IEEE, Nov. 2014, pp. 264–271. doi: 10.1109/ICTAI.2014.47.
- [59] L. Zheng, F. Chao, N. M. Parthaláin, D. Zhang, and Q. Shen, "Feature grouping and selection: A graph-based approach," *Inf. Sci.*, vol. 546, pp. 1256–1272, Feb. 2021, doi: 10.1016/j.ins.2020.09.022.
- [60] J. Wan, H. Chen, T. Li, B. Sang, and Z. Yuan, "Feature Grouping and Selection With Graph Theory in Robust Fuzzy Rough Approximation Space," *IEEE Trans. Fuzzy Syst.*, vol. 31, no. 1, pp. 213–225, Jan. 2023, doi: 10.1109/TFUZZ.2022.3185285.
- [61] M. García-Torres, F. Gómez-Vela, B. Melián-Batista, and J. M. Moreno-Vega, "High-dimensional feature selection via feature grouping: A Variable Neighborhood Search approach," *Inf. Sci.*, vol. 326, pp. 102–118, Jan. 2016, doi: 10.1016/j.ins.2015.07.041.
- [62] M. García-Torres, F. Gómez-Vela, F. Divina, D. P. Pinto-Roa, J. L. V. Noguera, and J. C. M. Román, "Scatter search for high-dimensional feature selection using feature grouping," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, Lille France: ACM, Jul. 2021, pp. 149–150. doi: 10.1145/3449726.3459481.
- [63] X.-F. Song, Y. Zhang, D.-W. Gong, and X.-Z. Gao, "A Fast Hybrid Feature Selection Based on Correlation-Guided Clustering and Particle Swarm

- Optimization for High-Dimensional Data,” *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9573–9586, Sep. 2022, doi: 10.1109/TCYB.2021.3061152.
- [64] M. García-Torres, R. Ruiz, and F. Divina, “Evolutionary feature selection on high dimensional data using a search space reduction approach,” *Eng. Appl. Artif. Intell.*, vol. 117, p. 105556, Jan. 2023, doi: 10.1016/j.engappai.2022.105556.
 - [65] S. Loscalzo, L. Yu, and C. Ding, “Consensus group stable feature selection,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*, Paris, France: ACM Press, 2009, p. 567. doi: 10.1145/1557019.1557084.
 - [66] Q. Xiao, H. Li, J. Tian, and Z. Wang, “Group-Wise Feature Selection for Supervised Learning,” in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, Singapore: IEEE, May 2022, pp. 3149–3153. doi: 10.1109/ICASSP43922.2022.9746666.
 - [67] A. Masoomi, C. Wu, T. Zhao, Z. Wang, P. Castaldi, and J. Dy, “Instance-wise feature grouping,” *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 13374–13386, 2020.
 - [68] F. Moslehi and A. Haeri, “A novel feature selection approach based on clustering algorithm,” *J. Stat. Comput. Simul.*, vol. 91, no. 3, pp. 581–604, Feb. 2021, doi: 10.1080/00949655.2020.1822358.
 - [69] M. Yousef, S. Jung, L. C. Showe, and M. K. Showe, “Recursive Cluster Elimination (RCE) for classification and feature selection from gene expression data,” *BMC Bioinformatics*, vol. 8, no. 1, p. 144, Dec. 2007, doi: 10.1186/1471-2105-8-144.
 - [70] P. Mitra, C. A. Murthy, and S. K. Pal, “Unsupervised feature selection using feature similarity,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 301–312, Mar. 2002, doi: 10.1109/34.990133.
 - [71] Guangrong Li, Xiaohua Hu, Xiajiong Shen, Xin Chen, and Zhoujun Li, “A novel unsupervised feature selection method for bioinformatics data sets through feature clustering,” in *2008 IEEE International Conference on Granular Computing*, Hangzhou: IEEE, Aug. 2008, pp. 41–47. doi: 10.1109/GRC.2008.4664788.
 - [72] T. F. Covões, E. R. Hruschka, L. N. de Castro, and Á. M. Santos, “A Cluster-Based Feature Selection Approach,” in *Hybrid Artificial Intelligence Systems*, vol. 5572, E. Corchado, X. Wu, E. Oja, Á. Herrero, and B. Baroque, Eds., in *Lecture Notes in Computer Science*, vol. 5572, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 169–176. doi: 10.1007/978-3-642-02319-4_20.
 - [73] X. Zhao, W. Deng, and Y. Shi, “Feature Selection with Attributes Clustering by Maximal Information Coefficient,” *Procedia Comput. Sci.*, vol. 17, pp. 70–79, 2013, doi: 10.1016/j.procs.2013.05.011.
 - [74] P.-Y. Zhou and K. C. C. Chan, “An unsupervised attribute clustering algorithm for unsupervised feature selection,” in *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Campus des Cordeliers, Paris, France: IEEE, Oct. 2015, pp. 1–7. doi: 10.1109/DSAA.2015.7344857.

- [75] X. Zhu, Y. Wang, Y. Li, Y. Tan, G. Wang, and Q. Song, “A new unsupervised feature selection algorithm using similarity-based feature clustering,” *Comput. Intell.*, vol. 35, no. 1, pp. 2–22, 2019, doi: 10.1111/coin.12192.
- [76] Z. Manbari, F. AkhlaghianTab, and C. Salavati, “Hybrid fast unsupervised feature selection for high-dimensional data,” *Expert Syst. Appl.*, vol. 124, pp. 97–118, Jun. 2019, doi: 10.1016/j.eswa.2019.01.016.
- [77] A. Yuan, J. Huang, C. Wei, W. Zhang, N. Zhang, and M. You, “Unsupervised Feature Selection via Feature-Grouping and Orthogonal Constraint,” in *2022 26th International Conference on Pattern Recognition (ICPR)*, Aug. 2022, pp. 720–726. doi: 10.1109/ICPR56361.2022.9956408.
- [78] I. Quinzán, J. M. Sotoca, and F. Pla, “Clustering-Based Feature Selection in Semi-supervised Problems,” in *2009 Ninth International Conference on Intelligent Systems Design and Applications*, Pisa, Italy: IEEE, 2009, pp. 535–540. doi: 10.1109/ISDA.2009.211.
- [79] Z. Song, X. Yang, Z. Xu, and I. King, “Graph-Based Semi-Supervised Learning: A Comprehensive Review,” *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–21, 2022, doi: 10.1109/TNNLS.2022.3155478.
- [80] G. Kostopoulos, S. Karlos, S. Kotsiantis, and O. Ragos, “Semi-supervised regression: A recent review,” *J. Intell. Fuzzy Syst.*, vol. 35, no. 2, pp. 1483–1500, Aug. 2018, doi: 10.3233/JIFS-169689.
- [81] R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki, “A Survey on semi-supervised feature selection methods,” *Pattern Recognit.*, vol. 64, pp. 141–158, Apr. 2017, doi: 10.1016/j.patcog.2016.11.003.
- [82] R. Bellazzi and B. Zupan, “Towards knowledge-based gene expression data mining,” *J. Biomed. Inform.*, vol. 40, no. 6, pp. 787–802, Dec. 2007, doi: 10.1016/j.jbi.2007.06.005.
- [83] C. Perscheid, “Integrative biomarker detection on high-dimensional gene expression data sets: a survey on prior knowledge approaches,” *Brief. Bioinform.*, vol. 22, no. 3, p. bbaa151, May 2021, doi: 10.1093/bib/bbaa151.
- [84] M. Yousef, A. Kumar, and B. Bakir-Gungor, “Application of Biological Domain Knowledge Based Feature Selection on Gene Expression Data,” *Entropy*, vol. 23, no. 1, p. 2, Dec. 2020, doi: 10.3390/e23010002.
- [85] M. Yousef and D. Voskergian, “TextNetTopics: Text Classification Based Word Grouping as Topics and Topics’ Scoring,” *Front. Genet.*, vol. 13, p. 893378, Jun. 2022, doi: 10.3389/fgene.2022.893378.
- [86] M. Ashburner *et al.*, “Gene Ontology: tool for the unification of biology,” *Nat. Genet.*, vol. 25, no. 1, pp. 25–29, May 2000, doi: 10.1038/75556.
- [87] M. Kanehisa, “KEGG: Kyoto Encyclopedia of Genes and Genomes,” *Nucleic Acids Res.*, vol. 28, no. 1, pp. 27–30, Jan. 2000, doi: 10.1093/nar/28.1.27.
- [88] M. Yousef, M. Ketany, L. Manevitz, L. C. Showe, and M. K. Showe, “Classification and biomarker identification using gene network modules and

- support vector machines,” *BMC Bioinformatics*, vol. 10, no. 1, p. 337, Dec. 2009, doi: 10.1186/1471-2105-10-337.
- [89] J. Wang *et al.*, “VISDA: an open-source caBIGTM analytical tool for data clustering and beyond,” *Bioinformatics*, vol. 23, no. 15, pp. 2024–2027, Aug. 2007, doi: 10.1093/bioinformatics/btm290.
 - [90] J. Qi and J. Tang, “Integrating gene ontology into discriminative powers of genes for feature selection in microarray data,” in *Proceedings of the 2007 ACM symposium on Applied computing - SAC '07*, Seoul, Korea: ACM Press, 2007, p. 430. doi: 10.1145/1244002.1244101.
 - [91] Y. Li, F.-X. Wu, and A. Ngom, “A review on machine learning principles for multi-view biological data integration,” *Brief. Bioinform.*, p. bbw113, Dec. 2016, doi: 10.1093/bib/bbw113.
 - [92] S. Falcon and R. Gentleman, “Using GOSTats to test gene lists for GO term association,” *Bioinformatics*, vol. 23, no. 2, pp. 257–258, Jan. 2007, doi: 10.1093/bioinformatics/btl567.
 - [93] G. Papachristoudis, S. Diplaris, and P. A. Mitkas, “SoFoCles: Feature filtering for microarray classification based on Gene Ontology,” *J. Biomed. Inform.*, vol. 43, no. 1, pp. 1–14, Feb. 2010, doi: 10.1016/j.jbi.2009.06.002.
 - [94] S. Mitra and S. Ghosh, “Feature Selection and Clustering of Gene Expression Profiles Using Biological Knowledge,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 42, no. 6, pp. 1590–1599, Nov. 2012, doi: 10.1109/TSMCC.2012.2209416.
 - [95] S. Ghosh and S. Mitra, “Gene selection using biological knowledge and fuzzy clustering,” in *2012 IEEE International Conference on Fuzzy Systems*, Brisbane, Australia: IEEE, Jun. 2012, pp. 1–9. doi: 10.1109/FUZZ-IEEE.2012.6250797.
 - [96] O. H. Fang, N. Mustapha, and Md. N. Sulaiman, “An integrative gene selection with association analysis for microarray data classification,” *Intell. Data Anal.*, vol. 18, no. 4, pp. 739–758, Jun. 2014, doi: 10.3233/IDA-140666.
 - [97] V. K. Raghu, X. Ge, P. K. Chrysanthi, and P. V. Benos, “Integrated Theory-and Data-Driven Feature Selection in Gene Expression Data Analysis,” in *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, San Diego, CA, USA: IEEE, Apr. 2017, pp. 1525–1532. doi: 10.1109/ICDE.2017.223.
 - [98] J. Piñero *et al.*, “The DisGeNET knowledge platform for disease genomics: 2019 update,” *Nucleic Acids Res.*, p. gkz1021, Nov. 2019, doi: 10.1093/nar/gkz1021.
 - [99] M. Yousef, L. Abdallah, and J. Allmer, “maTE: discovering expressed interactions between microRNAs and their targets,” *Bioinformatics*, vol. 35, no. 20, pp. 4020–4028, Oct. 2019, doi: 10.1093/bioinformatics/btz204.
 - [100] M. Yousef, J. Allmer, Y. İnal, and B. B. Gungor, “G-S-M: A Comprehensive Framework for Integrative Feature Selection in Omics Data Analysis and Beyond,” Apr. 01, 2024. doi: 10.1101/2024.03.30.585514.
 - [101] M. Yousef, E. Ülgen, and O. Uğur Sezerman, “CogNet: classification of gene expression data based on ranked active-subnetwork-oriented KEGG pathway

- enrichment analysis,” *PeerJ Comput. Sci.*, vol. 7, p. e336, Feb. 2021, doi: 10.7717/peerj-cs.336.
- [102] E. Ulgen, O. Ozisik, and O. U. Sezerman, “pathfindR: An R Package for Comprehensive Identification of Enriched Pathways in Omics Data Through Active Subnetworks,” *Front. Genet.*, vol. 10, p. 858, Sep. 2019, doi: 10.3389/fgene.2019.00858.
- [103] M. Yousef, G. Goy, R. Mitra, C. M. Eischen, A. Jabeer, and B. Bakir-Gungor, “miRcorrNet: machine learning-based integration of miRNA and mRNA expression profiles, combined with feature grouping and ranking,” *PeerJ*, vol. 9, p. e11458, May 2021, doi: 10.7717/peerj.11458.
- [104] M. Yousef, G. Goy, and B. Bakir-Gungor, “miRModuleNet: Detecting miRNA-mRNA Regulatory Modules,” *Front. Genet.*, vol. 13, p. 767455, Apr. 2022, doi: 10.3389/fgene.2022.767455.
- [105] M. Yousef, A. Sayıcı, and B. Bakir-Gungor, “Integrating Gene Ontology Based Grouping and Ranking into the Machine Learning Algorithm for Gene Expression Data Analysis,” in *Database and Expert Systems Applications - DEXA 2021 Workshops*, vol. 1479, G. Kotsis, A. M. Tjoa, I. Khalil, B. Moser, A. Mashkoor, J. Sametinger, A. Fensel, J. Martinez-Gil, L. Fischer, G. Czech, F. Sobieczky, and S. Khan, Eds., in *Communications in Computer and Information Science*, vol. 1479, Cham: Springer International Publishing, 2021, pp. 205–214. doi: 10.1007/978-3-030-87101-7_20.
- [106] M. Yousef, F. Ozdemir, A. Jaber, J. Allmer, and B. Bakir-Gungor, “PriPath: identifying dysregulated pathways from differential gene expression via grouping, scoring, and modeling with an embedded feature selection approach,” *BMC Bioinformatics*, vol. 24, no. 1, p. 60, Feb. 2023, doi: 10.1186/s12859-023-05187-2.
- [107] E. Qumsiyeh, L. Showe, and M. Yousef, “GediNET for discovering gene associations across diseases using knowledge based machine learning approach,” *Sci. Rep.*, vol. 12, no. 1, p. 19955, Nov. 2022, doi: 10.1038/s41598-022-24421-0.
- [108] M. Unlu Yazici, J. S. Marron, B. Bakir-Gungor, F. Zou, and M. Yousef, “Invention of 3Mint for feature grouping and scoring in multi-omics,” *Front. Genet.*, vol. 14, p. 1093326, Mar. 2023, doi: 10.3389/fgene.2023.1093326.
- [109] A. Jabeer, M. Temiz, B. Bakir-Gungor, and M. Yousef, “miRdisNET: Discovering microRNA biomarkers that are associated with diseases utilizing biological knowledge-based machine learning,” *Front. Genet.*, vol. 13, p. 1076554, Jan. 2023, doi: 10.3389/fgene.2022.1076554.
- [110] Y. Zhang, X. Lin, Z. Gao, and S. Bai, “A novel method for feature selection based on molecular interactive effect network,” *J. Pharm. Biomed. Anal.*, vol. 218, p. 114873, Sep. 2022, doi: 10.1016/j.jpba.2022.114873.
- [111] C. Perscheid, B. Grasnack, and M. Uflacker, “Integrative Gene Selection on Gene Expression Data: Providing Biological Context to Traditional Approaches,” *J. Integr. Bioinforma.*, vol. 16, no. 1, Dec. 2018, doi: 10.1515/jib-2018-0064.

- [112] C. Kuzudisli, B. Bakir-Gungor, B. F. Qaqish, and Malik. Yousef, “Effect of Recursive Cluster Elimination with Different Clustering Algorithms Applied to Gene Expression Data,” in *2023 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Sivas, Turkiye: IEEE, Oct. 2023, pp. 1–4. doi: 10.1109/ASYU58738.2023.10296734.
- [113] Y. Tang, Y.-Q. Zhang, Z. Huang, X. Hu, and Y. Zhao, “Recursive fuzzy granulation for gene subsets extraction and cancer classification,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 12, no. 6, pp. 723–730, 2008.
- [114] W. Du, Y. Sun, Y. Wang, Z. Cao, C. Zhang, and Y. Liang, “A novel multi-stage feature selection method for microarray expression data analysis,” *Int. J. Data Min. Bioinforma.*, vol. 7, no. 1, p. 58, 2013, doi: 10.1504/IJDMB.2013.050977.
- [115] C. Huang, “Feature Selection and Feature Stability Measurement Method for High-Dimensional Small Sample Data Based on Big Data Technology,” *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–12, Sep. 2021, doi: 10.1155/2021/3597051.
- [116] Y. Wang *et al.*, “Random Bits Forest: a Strong Classifier/Regressor for Big Data,” *Sci. Rep.*, vol. 6, no. 1, p. 30086, Jul. 2016, doi: 10.1038/srep30086.
- [117] L. Palaniyappan *et al.*, “Effective connectivity within a triple network brain system discriminates schizophrenia spectrum disorders from psychotic bipolar disorder at the single-subject level,” *Schizophr. Res.*, vol. 214, pp. 24–33, Dec. 2019, doi: 10.1016/j.schres.2018.01.006.
- [118] P. Lanka, D. Rangaprakash, M. N. Dretsch, J. S. Katz, T. S. Denney, and G. Deshpande, “Supervised machine learning for diagnostic classification from large-scale neuroimaging datasets,” *Brain Imaging Behav.*, vol. 14, no. 6, pp. 2378–2416, Dec. 2020, doi: 10.1007/s11682-019-00191-8.
- [119] P. Karunakaran, Y. B. Hamdan, and others, “Early prediction of autism spectrum disorder by computational approaches to fMRI analysis with early learning technique,” *J. Artif. Intell.*, vol. 2, no. 04, pp. 207–216, 2020.
- [120] D. C. Weis, D. P. Visco, and J.-L. Faulon, “Data mining PubChem using a support vector machine with the Signature molecular descriptor: Classification of factor Xla inhibitors,” *J. Mol. Graph. Model.*, vol. 27, no. 4, pp. 466–475, Nov. 2008, doi: 10.1016/j.jmglm.2008.08.004.
- [121] L.-K. Luo, D.-F. Huang, L.-J. Ye, Q.-F. Zhou, G.-F. Shao, and H. Peng, “Improving the computational efficiency of recursive cluster elimination for gene selection,” *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 8, no. 1, pp. 122–129, 2010.
- [122] D. Rangaprakash *et al.*, “Compromised hippocampus-striatum pathway as a potential imaging biomarker of mild-traumatic brain injury and posttraumatic stress disorder,” *Hum. Brain Mapp.*, vol. 38, no. 6, pp. 2843–2864, Jun. 2017, doi: 10.1002/hbm.23551.
- [123] C. Jin *et al.*, “Dynamic brain connectivity is a better predictor of PTSD than static connectivity: Dynamic Brain Connectivity,” *Hum. Brain Mapp.*, vol. 38, no. 9, pp. 4479–4496, Sep. 2017, doi: 10.1002/hbm.23676.

- [124] X. Zhao, L. Wang, and G. Chen, "Joint Covariate Detection on Expression Profiles for Identifying MicroRNAs Related to Venous Metastasis in Hepatocellular Carcinoma," *Sci. Rep.*, vol. 7, no. 1, p. 5349, Dec. 2017, doi: 10.1038/s41598-017-05776-1.
- [125] N. Chaitra, P. A. Vijaya, and G. Deshpande, "Diagnostic prediction of autism spectrum disorder using complex network measures in a machine learning framework," *Biomed. Signal Process. Control*, vol. 62, p. 102099, Sep. 2020, doi: 10.1016/j.bspc.2020.102099.
- [126] N. S. Ersoz, B. Bakir-Gungor, and M. Yousef, "GeNetOntology: identifying affected gene ontology terms via grouping, scoring, and modeling of gene expression data utilizing biological knowledge-based machine learning," *Front. Genet.*, vol. 14, p. 1139082, Aug. 2023, doi: 10.3389/fgene.2023.1139082.
- [127] D. Voskergian, B. Bakir-Gungor, and M. Yousef, "TextNetTopics Pro, a topic model-based text classification for short text by integration of semantic and document-topic distribution information," *Front. Genet.*, vol. 14, p. 1243874, Oct. 2023, doi: 10.3389/fgene.2023.1243874.
- [128] B. Bakir-Gungor, M. Temiz, A. Jabeer, D. Wu, and M. Yousef, "microBiomeGSM: the identification of taxonomic biomarkers from metagenomic data using grouping, scoring and modeling (G-S-M) approach," *Front. Microbiol.*, vol. 14, p. 1264941, Nov. 2023, doi: 10.3389/fmicb.2023.1264941.
- [129] E. Qumsiyeh, Z. Salah, and M. Yousef, "miRGediNET: A comprehensive examination of common genes in miRNA-Target interactions and disease associations: Insights from a grouping-scoring-modeling approach," *Heliyon*, vol. 9, no. 12, p. e22666, Dec. 2023, doi: 10.1016/j.heliyon.2023.e22666.
- [130] Ü. G. Söylemez, M. Yousef, and B. Bakir-Gungor, "AMP-GSM: Prediction of Antimicrobial Peptides via a Grouping–Scoring–Modeling Approach," *Appl. Sci.*, vol. 13, no. 8, p. 5106, Apr. 2023, doi: 10.3390/app13085106.
- [131] S. Na, L. Xumin, and G. Yong, "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm," in *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, Jian, China: IEEE, Apr. 2010, pp. 63–67. doi: 10.1109/IITSI.2010.74.
- [132] A. Bhat, "K-Medoids Clustering Using Partitioning Around Medoids for Performing Face Recognition," *Int. J. Soft Comput. Math. Control*, vol. 3, no. 3, pp. 1–12, Aug. 2014, doi: 10.14810/ijscmc.2014.3301.
- [133] C. F. Olson, "Parallel algorithms for hierarchical clustering," *Parallel Comput.*, vol. 21, no. 8, pp. 1313–1325, Aug. 1995, doi: 10.1016/0167-8191(95)00017-I.
- [134] T. Li, A. Rezaeipannah, and E. M. Tag El Din, "An ensemble agglomerative hierarchical clustering algorithm based on clusters clustering technique and the novel similarity measurement," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 3828–3842, Jun. 2022, doi: 10.1016/j.jksuci.2022.04.010.
- [135] A. Habib, M. Akram, and C. Kahraman, "Minimum spanning tree hierarchical clustering algorithm: A new Pythagorean fuzzy similarity measure for the analysis

- of functional brain networks,” *Expert Syst. Appl.*, vol. 201, p. 117016, Sep. 2022, doi: 10.1016/j.eswa.2022.117016.
- [136] C. M. Bishop, *Pattern recognition and machine learning*. in Information science and statistics. New York: Springer, 2006.
 - [137] E. Clough and T. Barrett, “The Gene Expression Omnibus Database,” in *Statistical Genomics*, vol. 1418, E. Mathé and S. Davis, Eds., in Methods in Molecular Biology, vol. 1418. , New York, NY: Springer New York, 2016, pp. 93–110. doi: 10.1007/978-1-4939-3578-9_5.
 - [138] M. R. Berthold *et al.*, “KNIME - the Konstanz information miner: version 2.0 and beyond,” *ACM SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 26–31, Nov. 2009, doi: 10.1145/1656274.1656280.
 - [139] H.-S. Park and C.-H. Jun, “A simple and fast algorithm for K-medoids clustering,” *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3336–3341, Mar. 2009, doi: 10.1016/j.eswa.2008.01.039.
 - [140] T. Barrett *et al.*, “NCBI GEO: archive for functional genomics data sets—update,” *Nucleic Acids Res.*, vol. 41, no. D1, pp. D991–D995, Nov. 2012, doi: 10.1093/nar/gks1193.
 - [141] K. Tomczak, P. Czerwińska, and M. Wiznerowicz, “Review The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge,” *Współczesna Onkol.*, vol. 1A, pp. 68–77, 2015, doi: 10.5114/wo.2014.47136.
 - [142] M. J. Goldman *et al.*, “Visualizing and interpreting cancer genomics data via the Xena platform,” *Nat. Biotechnol.*, vol. 38, no. 6, pp. 675–678, Jun. 2020, doi: 10.1038/s41587-020-0546-8.
 - [143] F. Beghini *et al.*, “Integrating taxonomic, functional, and strain-level profiling of diverse microbial communities with bioBakery 3,” *eLife*, vol. 10, p. e65088, May 2021, doi: 10.7554/eLife.65088.
 - [144] C. Kuzudisli, B. Bakir-Gungor, B. Qaqish, and M. Yousef, “RCE-IFE: Recursive Cluster Elimination with Intra-cluster Feature Elimination,” *bioRxiv*, 2024, doi: 10.1101/2024.02.28.580487.
 - [145] G. Zeller *et al.*, “Potential of fecal microbiota for early-stage detection of colorectal cancer,” *Mol. Syst. Biol.*, vol. 10, no. 11, p. 766, Nov. 2014, doi: 10.15252/msb.20145645.
 - [146] MetaHIT Consortium *et al.*, “A human gut microbial gene catalogue established by metagenomic sequencing,” *Nature*, vol. 464, no. 7285, pp. 59–65, Mar. 2010, doi: 10.1038/nature08821.
 - [147] J. Qin *et al.*, “A metagenome-wide association study of gut microbiota in type 2 diabetes,” *Nature*, vol. 490, no. 7418, pp. 55–60, Oct. 2012, doi: 10.1038/nature11450.
 - [148] L. J. van ’t Veer *et al.*, “Gene expression profiling predicts clinical outcome of breast cancer,” *Nature*, vol. 415, no. 6871, pp. 530–536, Jan. 2002, doi: 10.1038/415530a.

- [149] M. A. Shipp *et al.*, “Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning,” *Nat. Med.*, vol. 8, no. 1, pp. 68–74, Jan. 2002, doi: 10.1038/nm0102-68.
- [150] T. R. Golub *et al.*, “Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring,” *Science*, vol. 286, no. 5439, pp. 531–537, Oct. 1999, doi: 10.1126/science.286.5439.531.
- [151] D. Singh *et al.*, “Gene expression correlates of clinical prostate cancer behavior,” *Cancer Cell*, vol. 1, no. 2, pp. 203–209, Mar. 2002, doi: 10.1016/S1535-6108(02)00030-2.
- [152] S. Prusty, S. Patnaik, and S. K. Dash, “SKCV: Stratified K-fold cross-validation on ML classifiers for predicting cervical cancer,” *Front. Nanotechnol.*, vol. 4, p. 972421, Aug. 2022, doi: 10.3389/fnano.2022.972421.
- [153] M. L. McHugh, “Interrater reliability: the kappa statistic,” *Biochem. Medica*, pp. 276–282, 2012, doi: 10.11613/BM.2012.031.
- [154] Hanchuan Peng, Fuhui Long, and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005, doi: 10.1109/TPAMI.2005.159.
- [155] L. Yu and H. Liu, “Feature selection for high-dimensional data: A fast correlation-based filter solution,” in *Proceedings of the 20th international conference on machine learning (ICML-03)*, 2003, pp. 856–863.
- [156] F. Fleuret, “Fast binary feature selection with conditional mutual information,” *J. Mach. Learn. Res.*, vol. 5, no. 9, 2004.
- [157] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA: ACM, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.
- [158] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
- [159] E. A. Frick *et al.*, “MicroRNA-190b Targets RFWD3 in Estrogen Receptor–Positive Breast Cancer,” *Breast Cancer Basic Clin. Res.*, vol. 18, p. 11782234241234771, Jan. 2024, doi: 10.1177/11782234241234771.
- [160] W. Dai *et al.*, “miR-148b-3p, miR-190b, and miR-429 Regulate Cell Progression and Act as Potential Biomarkers for Breast Cancer,” *J. Breast Cancer*, vol. 22, no. 2, p. 219, 2019, doi: 10.4048/jbc.2019.22.e19.
- [161] Y. Yu *et al.*, “miR-190 suppresses breast cancer metastasis by regulation of TGF- β -induced epithelial–mesenchymal transition,” *Mol. Cancer*, vol. 17, no. 1, p. 70, Dec. 2018, doi: 10.1186/s12943-018-0818-9.
- [162] K. Hua *et al.*, “miR-135b, upregulated in breast cancer, promotes cell growth and disrupts the cell cycle by regulating LATS2,” *Int. J. Oncol.*, vol. 48, no. 5, pp. 1997–2006, May 2016, doi: 10.3892/ijo.2016.3405.

- [163] T. H. Vo *et al.*, “miRNAs as biomarkers of therapeutic response to HER2-targeted treatment in breast cancer: A systematic review,” *Biochem. Biophys. Rep.*, vol. 37, p. 101588, Mar. 2024, doi: 10.1016/j.bbrep.2023.101588.
- [164] N. G. Egeland *et al.*, “MiR-18a and miR-18b are expressed in the stroma of oestrogen receptor alpha negative breast cancers,” *BMC Cancer*, vol. 20, no. 1, p. 377, Dec. 2020, doi: 10.1186/s12885-020-06857-7.
- [165] K. Jonsdottir *et al.*, “Validation of Expression Patterns for Nine miRNAs in 204 Lymph-Node Negative Breast Cancers,” *PLoS ONE*, vol. 7, no. 11, p. e48692, Nov. 2012, doi: 10.1371/journal.pone.0048692.
- [166] J. Wang, H. Liu, and M. Li, “Downregulation of miR-505 promotes cell proliferation, migration and invasion, and predicts poor prognosis in breast cancer,” *Oncol. Lett.*, May 2019, doi: 10.3892/ol.2019.10334.
- [167] Y. Lu, X. Hu, and X. Yang, “miR-934 promotes breast cancer metastasis by regulation of PTEN and epithelial–mesenchymal transition,” *Tissue Cell*, vol. 71, p. 101581, Aug. 2021, doi: 10.1016/j.tice.2021.101581.
- [168] J. A. Contreras-Rodríguez, J. Puente-Rivera, D. M. Córdova-Esparza, S. I. Nuñez-Olvera, and M. B. Silva-Cázares, “Bioinformatic miRNA-mRNAs Analysis Reveals to miR-934 as a Potential Regulator of the Epithelial–Mesenchymal Transition in Triple-Negative Breast Cancer,” *Cells*, vol. 12, no. 6, p. 834, Mar. 2023, doi: 10.3390/cells12060834.
- [169] D. Szklarczyk *et al.*, “STRING v10: protein–protein interaction networks, integrated over the tree of life,” *Nucleic Acids Res.*, vol. 43, no. D1, pp. D447–D452, Jan. 2015, doi: 10.1093/nar/gku1003.
- [170] D. Otasek, J. H. Morris, J. Bouças, A. R. Pico, and B. Demchak, “Cytoscape Automation: empowering workflow-based network analysis,” *Genome Biol.*, vol. 20, no. 1, Dec. 2019, doi: 10.1186/s13059-019-1758-4.
- [171] R. Abou Ziki, Y. Luo, V. Vlaeminck-Guillem, M. Le Romancer, and C. Zhang, “Involvement of the MEN1 Gene in Hormone-Related Cancers: Clues from Molecular Studies, Mouse Models, and Patient Investigations,” *Endocrines*, vol. 1, no. 2, pp. 58–81, Oct. 2020, doi: 10.3390/endocrines1020007.
- [172] L. Xin, “EZH2 accompanies prostate cancer progression,” *Nat. Cell Biol.*, vol. 23, no. 9, pp. 934–936, Sep. 2021, doi: 10.1038/s41556-021-00744-4.

APPENDIX

Table A.1 Comparative performance evaluation of RCE-IFE with SVM-RCE: Mean Accuracy and Sensitivity on gene expression, miRNA and methylation datasets

Dataset	Accuracy			Sensitivity		
	SVM-RCE	RCE-IFE-SVM	RCE-IFE	SVM-RCE	RCE-IFE-SVM	RCE-IFE
GDS1962	0.93 ± 0.10	0.94 ± 0.09	0.93 ± 0.08	0.93 ± 0.12	0.95 ± 0.09	0.94 ± 0.10
GDS2519	0.50 ± 0.15	0.55 ± 0.14	0.52 ± 0.15	0.51 ± 0.24	0.56 ± 0.22	0.52 ± 0.25
GDS2547	0.73 ± 0.10	0.74 ± 0.11	0.73 ± 0.10	0.74 ± 0.15	0.77 ± 0.15	0.74 ± 0.17
GDS2609	0.92 ± 0.16	0.91 ± 0.18	0.96 ± 0.14	0.91 ± 0.21	0.88 ± 0.25	0.97 ± 0.16
GDS3268	0.72 ± 0.11	0.71 ± 0.12	0.69 ± 0.12	0.70 ± 0.17	0.70 ± 0.19	0.73 ± 0.16
GDS3646	0.73 ± 0.14	0.71 ± 0.16	0.69 ± 0.14	0.88 ± 0.17	0.81 ± 0.19	0.81 ± 0.16
GDS3794	0.88 ± 0.15	0.87 ± 0.15	0.87 ± 0.16	0.87 ± 0.24	0.85 ± 0.24	0.86 ± 0.25
GDS3837	0.95 ± 0.06	0.94 ± 0.06	0.95 ± 0.06	0.94 ± 0.09	0.94 ± 0.09	0.96 ± 0.08
GDS3874	0.71 ± 0.14	0.70 ± 0.17	0.79 ± 0.13	0.83 ± 0.17	0.81 ± 0.18	0.92 ± 0.13
GDS3875	0.78 ± 0.14	0.79 ± 0.14	0.76 ± 0.15	0.85 ± 0.16	0.88 ± 0.15	0.88 ± 0.16
GDS3929	0.65 ± 0.08	0.63 ± 0.09	0.59 ± 0.16	0.95 ± 0.12	0.93 ± 0.14	0.79 ± 0.22
GDS4228	0.66 ± 0.09	0.64 ± 0.07	0.74 ± 0.15	0.96 ± 0.12	0.94 ± 0.13	0.88 ± 0.16
GDS4824	0.86 ± 0.26	0.89 ± 0.21	0.87 ± 0.27	0.83 ± 0.38	0.85 ± 0.36	0.83 ± 0.38
GDS5037	0.72 ± 0.16	0.72 ± 0.16	0.77 ± 0.12	0.80 ± 0.18	0.81 ± 0.18	0.89 ± 0.13
GDS5093	0.79 ± 0.21	0.80 ± 0.21	0.81 ± 0.21	0.85 ± 0.25	0.83 ± 0.25	0.88 ± 0.23
GDS5499	0.91 ± 0.07	0.92 ± 0.07	0.89 ± 0.09	0.94 ± 0.07	0.97 ± 0.06	0.94 ± 0.08
GSE157103	0.88 ± 0.11	0.91 ± 0.10	0.86 ± 0.13	0.88 ± 0.14	0.94 ± 0.10	0.89 ± 0.16
TCGA-BLCA.methylation450	0.90 ± 0.11	0.94 ± 0.09	0.92 ± 0.12	0.94 ± 0.11	0.95 ± 0.10	0.93 ± 0.13
TCGA-BLCA.mirna	0.94 ± 0.10	0.92 ± 0.11	0.92 ± 0.10	0.94 ± 0.11	0.93 ± 0.14	0.94 ± 0.12
TCGA-BRCA.methylation450	0.95 ± 0.06	0.95 ± 0.06	0.97 ± 0.05	0.91 ± 0.14	0.91 ± 0.15	0.95 ± 0.11
Average	0.81 ± 0.13	0.81 ± 0.12	0.81 ± 0.13	0.86 ± 0.17	0.86 ± 0.17	0.86 ± 0.17

Table A.2 Comparative performance evaluation of RCE-IFE with SVM-RCE: Mean Specificity and mean Precision on gene expression, miRNA and methylation datasets

Dataset	Specificity	Precision
---------	-------------	-----------

	SVM-RCE	RCE-IFE-SVM	RCE-IFE	SVM-RCE	RCE-IFE-SVM	RCE-IFE
GDS1962	0.93 ± 0.21	0.92 ± 0.18	0.92 ± 0.20	0.97 ± 0.07	0.97 ± 0.07	0.97 ± 0.07
GDS2519	0.50 ± 0.21	0.54 ± 0.22	0.51 ± 0.20	0.45 ± 0.18	0.52 ± 0.18	0.46 ± 0.17
GDS2547	0.71 ± 0.16	0.72 ± 0.16	0.71 ± 0.17	0.73 ± 0.12	0.75 ± 0.12	0.73 ± 0.13
GDS2609	0.93 ± 0.26	0.97 ± 0.17	0.94 ± 0.24	0.97 ± 0.09	0.99 ± 0.07	0.98 ± 0.09
GDS3268	0.73 ± 0.17	0.72 ± 0.19	0.64 ± 0.19	0.76 ± 0.13	0.76 ± 0.14	0.71 ± 0.12
GDS3646	0.38 ± 0.34	0.48 ± 0.36	0.37 ± 0.36	0.78 ± 0.13	0.81 ± 0.13	0.77 ± 0.12
GDS3794	0.90 ± 0.23	0.90 ± 0.23	0.88 ± 0.23	0.93 ± 0.15	0.93 ± 0.15	0.91 ± 0.17
GDS3837	0.95 ± 0.09	0.94 ± 0.09	0.94 ± 0.09	0.96 ± 0.07	0.95 ± 0.08	0.95 ± 0.08
GDS3874	0.52 ± 0.28	0.52 ± 0.32	0.58 ± 0.29	0.75 ± 0.13	0.75 ± 0.15	0.80 ± 0.13
GDS3875	0.67 ± 0.28	0.63 ± 0.27	0.56 ± 0.32	0.83 ± 0.14	0.81 ± 0.13	0.79 ± 0.14
GDS3929	0.07 ± 0.21	0.04 ± 0.15	0.18 ± 0.27	0.68 ± 0.07	0.66 ± 0.06	0.65 ± 0.13
GDS4228	0.07 ± 0.17	0.04 ± 0.13	0.45 ± 0.36	0.67 ± 0.05	0.66 ± 0.04	0.78 ± 0.13
GDS4824	0.88 ± 0.33	0.93 ± 0.26	0.90 ± 0.30	0.91 ± 0.23	0.96 ± 0.14	0.91 ± 0.25
GDS5037	0.51 ± 0.38	0.51 ± 0.34	0.47 ± 0.36	0.82 ± 0.14	0.81 ± 0.12	0.82 ± 0.11
GDS5093	0.68 ± 0.47	0.73 ± 0.45	0.69 ± 0.46	0.88 ± 0.18	0.90 ± 0.17	0.88 ± 0.18
GDS5499	0.84 ± 0.18	0.83 ± 0.20	0.77 ± 0.19	0.94 ± 0.07	0.93 ± 0.08	0.90 ± 0.07
GSE157103	0.87 ± 0.19	0.87 ± 0.21	0.80 ± 0.21	0.93 ± 0.10	0.93 ± 0.10	0.89 ± 0.11
TCGA-BLCA.methylation450	0.84 ± 0.26	0.90 ± 0.21	0.89 ± 0.22	0.94 ± 0.10	0.96 ± 0.08	0.95 ± 0.10
TCGA-BLCA.mirna	0.94 ± 0.18	0.89 ± 0.22	0.89 ± 0.22	0.97 ± 0.07	0.95 ± 0.09	0.95 ± 0.09
TCGA-BRCA.methylation450	0.97 ± 0.06	0.97 ± 0.06	0.98 ± 0.05	0.95 ± 0.09	0.96 ± 0.09	0.98 ± 0.07
Average	0.69 ± 0.23	0.70 ± 0.22	0.70 ± 0.25	0.84 ± 0.12	0.85 ± 0.11	0.84 ± 0.12

Table A.3 Comparative performance evaluation of RCE-IFE with SVM-RCE: Mean F-measure and mean Cohen’s kappa on gene expression, miRNA and methylation datasets

Dataset	F-measure			Cohen’s kappa		
	SVM-RCE	RCE-IFE-SVM	RCE-IFE	SVM-RCE	RCE-IFE-SVM	RCE-IFE
GDS1962	0.95 ± 0.08	0.96 ± 0.07	0.95 ± 0.06	0.83 ± 0.25	0.86 ± 0.21	0.84 ± 0.21
GDS2519	0.49 ± 0.15	0.53 ± 0.15	0.50 ± 0.17	0.002 ± 0.29	0.10 ± 0.28	0.03 ± 0.30
GDS2547	0.73 ± 0.11	0.75 ± 0.11	0.73 ± 0.12	0.45 ± 0.20	0.49 ± 0.22	0.46 ± 0.21
GDS2609	0.93 ± 0.13	0.93 ± 0.14	0.98 ± 0.08	0.83 ± 0.34	0.83 ± 0.33	0.90 ± 0.30
GDS3268	0.72 ± 0.13	0.71 ± 0.14	0.71 ± 0.12	0.43 ± 0.22	0.42 ± 0.24	0.37 ± 0.24
GDS3646	0.83 ± 0.09	0.79 ± 0.14	0.78 ± 0.11	0.26 ± 0.36	0.28 ± 0.38	0.17 ± 0.37

GDS3794	0.88 ± 0.15	0.87 ± 0.16	0.87 ± 0.17	0.76 ± 0.30	0.75 ± 0.31	0.73 ± 0.33
GDS3837	0.95 ± 0.06	0.94 ± 0.06	0.95 ± 0.06	0.89 ± 0.11	0.88 ± 0.13	0.90 ± 0.12
GDS3874	0.78 ± 0.12	0.77 ± 0.14	0.85 ± 0.10	0.35 ± 0.31	0.33 ± 0.37	0.51 ± 0.32
GDS3875	0.82 ± 0.12	0.84 ± 0.11	0.82 ± 0.12	0.52 ± 0.30	0.52 ± 0.31	0.46 ± 0.34
GDS3929	0.78 ± 0.07	0.77 ± 0.08	0.71 ± 0.14	0.01 ± 0.21	-0.03 ± 0.19	-0.03 ± 0.32
GDS4228	0.79 ± 0.07	0.77 ± 0.07	0.81 ± 0.11	0.02 ± 0.20	-0.03 ± 0.14	0.34 ± 0.38
GDS4824	0.96 ± 0.10	0.97 ± 0.09	0.98 ± 0.08	0.71 ± 0.52	0.78 ± 0.42	0.73 ± 0.55
GDS5037	0.80 ± 0.13	0.80 ± 0.13	0.85 ± 0.09	0.30 ± 0.40	0.31 ± 0.38	0.36 ± 0.36
GDS5093	0.85 ± 0.16	0.84 ± 0.17	0.86 ± 0.16	0.51 ± 0.50	0.53 ± 0.48	0.55 ± 0.51
GDS5499	0.94 ± 0.05	0.95 ± 0.05	0.92 ± 0.06	0.79 ± 0.18	0.81 ± 0.19	0.72 ± 0.21
GSE157103	0.89 ± 0.10	0.93 ± 0.08	0.88 ± 0.11	0.74 ± 0.23	0.81 ± 0.23	0.70 ± 0.26
TCGA- BLCA.meth ylation450	0.93 ± 0.08	0.95 ± 0.07	0.93 ± 0.10	0.77 ± 0.26	0.85 ± 0.22	0.81 ± 0.27
TCGA- BLCA.mirna	0.95 ± 0.08	0.93 ± 0.10	0.94 ± 0.08	0.86 ± 0.22	0.81 ± 0.25	0.82 ± 0.24
TCGA- BRCA.meth ylation450	0.92 ± 0.09	0.92 ± 0.10	0.96 ± 0.07	0.88 ± 0.14	0.89 ± 0.14	0.94 ± 0.11
Average	0.84 ± 0.10	0.85 ± 0.11	0.85 ± 0.11	0.55 ± 0.28	0.56 ± 0.27	0.57 ± 0.30

Table A.4 Comparative performance evaluation of RCE-IFE with SVM-RCE in different metrics on metagenomics datasets

Metric	Method	CRC_enzyme	CRC_pathway	CRC_species	CRC_species_II	IBD	IBDMDB	T2D	Average
Accuracy	RCE-IFE	0.69 ± 0.04	0.64 ± 0.04	0.74 ± 0.04	0.74 ± 0.14	0.79 ± 0.07	0.90 ± 0.03	0.62 ± 0.09	0.73 ± 0.06
	SVM-RCE	0.65 ± 0.04	0.64 ± 0.04	0.62 ± 0.05	0.51 ± 0.16	0.75 ± 0.07	0.74 ± 0.03	0.56 ± 0.09	0.64 ± 0.07
Sensitivity	RCE-IFE	0.66 ± 0.06	0.62 ± 0.06	0.66 ± 0.07	0.82 ± 0.18	0.82 ± 0.10	0.97 ± 0.02	0.63 ± 0.14	0.74 ± 0.09
	SVM-RCE	0.56 ± 0.06	0.60 ± 0.07	0.45 ± 0.12	0.51 ± 0.32	0.69 ± 0.11	0.94 ± 0.03	0.64 ± 0.22	0.63 ± 0.13
Specificity	RCE-IFE	0.71 ± 0.05	0.66 ± 0.06	0.82 ± 0.05	0.67 ± 0.23	0.75 ± 0.12	0.77 ± 0.07	0.62 ± 0.13	0.71 ± 0.10
	SVM-RCE	0.73 ± 0.06	0.68 ± 0.07	0.79 ± 0.10	0.52 ± 0.33	0.81 ± 0.10	0.35 ± 0.10	0.47 ± 0.20	0.62 ± 0.14
Precision	RCE-IFE	0.69 ± 0.05	0.64 ± 0.05	0.78 ± 0.05	0.74 ± 0.16	0.78 ± 0.09	0.90 ± 0.03	0.62 ± 0.09	0.74 ± 0.07

	SVM-RCE	0.68 ± 0.05	0.65 ± 0.05	0.69 ± 0.10	0.51 ± 0.24	0.79 ± 0.08	0.75 ± 0.03	0.54 ± 0.11	0.66 ± 0.09
F-measure	RCE-IFE	0.68 ± 0.05	0.63 ± 0.05	0.71 ± 0.05	0.76 ± 0.13	0.79 ± 0.07	0.93 ± 0.02	0.62 ± 0.10	0.73 ± 0.07
	SVM-RCE	0.61 ± 0.05	0.62 ± 0.05	0.53 ± 0.11	0.53 ± 0.18	0.73 ± 0.08	0.83 ± 0.02	0.58 ± 0.14	0.63 ± 0.09
Cohen's kappa	RCE-IFE	0.37 ± 0.08	0.28 ± 0.09	0.48 ± 0.07	0.49 ± 0.27	0.57 ± 0.15	0.77 ± 0.07	0.25 ± 0.18	0.46 ± 0.13
	SVM-RCE	0.30 ± 0.08	0.27 ± 0.08	0.24 ± 0.11	0.03 ± 0.31	0.50 ± 0.14	0.33 ± 0.10	0.11 ± 0.19	0.25 ± 0.14

Table A.5 Hyper parameter settings for different classifiers

Classifier	RandomizedSearchCV parameters
Adaboost	params['n_estimators'] = (10,25,50,75,100), n_iter=500, scoring='accuracy', n_jobs=1, cv=2, random_state=1
DT	params['criterion'] = ('gini', 'entropy'), n_iter=500, scoring='accuracy', n_jobs=1, cv=2, random_state=1
Logitboost	params['n_estimators'] = (10,25,50,75,100), n_iter=500, scoring='accuracy', n_jobs=1, cv=2, random_state=1
RF	params['n_estimators'] = (10,25,50,75,100), n_iter=500, scoring='accuracy', n_jobs=1, cv=2, random_state=1
SVC	params['C'] = (1e-6, 100.0, 'log-uniform'), params['gamma'] = (1e-6, 100.0, 'log-uniform'), params['degree'] = (1,5), params['kernel'] = ['linear']
Stacking (Logitboost + KNN)	final_estimator=RandomForestClassifier(),scoring='accuracy'
Stacking (SVC + KNN)	params['C'] = (1e-6, 100.0, 'log-uniform'), params['gamma'] = (1e-6, 100.0, 'log-uniform'), n_iter=500, scoring='accuracy', n_jobs=1, cv=2, random_state=1
XGBClassifier	params['n_estimators'] = (10,25,50,75,100), n_iter=500, scoring='accuracy', n_jobs=1, cv=2, random_state=1

CURRICULUM VITAE

2004 – 2006	B.Sc., Electrical and Electronic Engineering, Eastern Mediterranean University, Famagusta, NORTH CYPRUS
2006 – 2009	B.Sc., Electrical and Electronic Engineering, Gaziantep University, Gaziantep, TURKEY
2010– 2013	M.Sc., Software Engineering, Boğaziçi University, Istanbul, TURKEY
2018 – 2024	Doctoral Candidate, Electrical and Computer Engineering, Abdullah Gul University, Kayseri, TURKEY

SELECTED PUBLICATIONS AND PRESENTATIONS

J1) C. Kuzudisli, B. Bakir-Gungor, N. Bulut, B.Qaqish, M. Yousef, Review of feature selection approaches based on grouping of features published in PeerJ Life & Environment Journal (July. 2023).

J2) C. Kuzudisli, B. Bakir-Gungor, B.Qaqish, M. Yousef, RCE-IFE: Recursive Cluster Elimination with Intra-cluster Feature Elimination (accepted in PeerJ Computer Science Journal).

C1) C. Kuzudisli, B. Bakir-Gungor, B.Qaqish, M. Yousef, Effect of Recursive Cluster Elimination with Different Clustering Algorithms Applied to Gene Expression Data in Proc. of Innovations in Intelligent Systems and Applications Conference (Oct. 2023).