

**GRAPH-BASED FEATURE SELECTION MODEL FOR  
GENES' PHENOTYPE PREDICTION**

**CONSOLATA GAKII MUGWIKA**

**DOCTOR OF PHILOSOPHY**

**(Information Technology)**

**JOMO KENYATTA UNIVERSITY  
OF  
AGRICULTURE AND TECHNOLOGY**

**2022**

# **Graph-Based Feature Selection Model for Genes' Phenotype Prediction**

**Consolata Gakii Mugwika**

**A Thesis Submitted in Partial Fulfillment of the Requirements for the  
Degree of Doctor of Philosophy in Information Technology of the Jomo  
Kenyatta University of Agriculture and Technology**

**2022**

## DECLARATION

This thesis is my original work and has not been presented for a degree in any other University.

Signature.....Date.....

**Consolata Gakii Mugwika**

This thesis has been submitted for examination with our approval as the university supervisors.

Signature.....Date.....

**Dr. Richard Rimiru, PhD**  
**JKUAT, Kenya**

Signature.....Date.....

**Dr. Paul. O. Mireji, PhD**  
**BioRI-KALRO, Kenya**

## **DEDICATION**

This work is dedicated to my loving husband Mwirichia, our dear daughter Joy and our Dear son Joe for supporting me tirelessly during the entire PhD journey. Thank you very much.

## **ACKNOWLEDGEMENTS**

I thank God for granting me the opportunity to walk the PhD journey.

I utmost appreciate my supervisors Dr. Richard Rimiru and Dr. Paul O. Mireji, for their support, training, mentorship, and encouragement. God bless you abundantly.

I specially thank the entire faculty members in the School of Computing and Information Technology for their positive feedback during seminar presentations.

I thank Prof. Xin Gao (KAUST), Prof. Tom Freeman, Dr. Mattia Chiesa and Dr. Stefan Naulaerts; International scholars who created time from their busy schedule and responded to me with guidance when I reached out to them. Thank you very much.

## TABLE OF CONTENTS

<b>DECLARATION.....</b>	<b>ii</b>
<b>DEDICATION.....</b>	<b>iii</b>
<b>ACKNOWLEDGEMENTS.....</b>	<b>iv</b>
<b>ABBREVIATIONS AND ACRONYMS.....</b>	<b>xv</b>
<b>ABSTRACT.....</b>	<b>xix</b>
<b>CHAPTER ONE .....</b>	<b>1</b>
<b>INTRODUCTION.....</b>	<b>1</b>
1.1 Background of the study.....	1
1.1.1 Genes and gene expression.....	1
1.1.2 Gene phenotype .....	2
1.1.3 Sources of big data.....	2
1.1.4 Curse of dimensionality .....	3
1.1.5 Feature engineering.....	3
1.1.6 Processing performance .....	4
1.1.7 Dimensionality reduction.....	5
1.2 Problem statement .....	7
1.3 Main objective.....	8

1.4 Specific objectives.....	8
1.5 Justification .....	8
1.6 Scope of the study .....	9
1.7 Knowledge Contributions.....	9
1.7 Thesis organization.....	10
<b>CHAPTER TWO .....</b>	<b>11</b>
<b>LITERATURE REVIEW.....</b>	<b>11</b>
2.1 Introduction to bioinformatics.....	11
2.2 Data structures used in bioinformatics .....	12
2.2.1 Hash tables.....	13
2.2.2 Suffix tree .....	14
2.2.3 Suffix array .....	15
2.2.4 Burrows-Wheeler transform .....	15
2.3 Mapping algorithms .....	16
2.3.1 Algorithms based on hash table .....	16
2.3.2 Algorithms based on Burrows-Wheeler transform.....	18
2.3.3 Algorithm based on Suffix array .....	19
2.4 Feature counting .....	21

2.4.1 Count-based quantifiers .....	21
2.4.2 Poisson model-based quantifiers .....	21
2.5 Feature Selection Methods .....	22
2.5.1 Filter-based Methods .....	22
2.5.1.4 Minimum Redundancy Maximum Relevance (mRMR) .....	31
2.5.2 Wrapper-based Methods .....	34
2.5.3 Embedded methods.....	34
2.6 Feature Extraction methods.....	36
2.6.1 Principal Component Analysis (PCA).....	36
2.6.2 Partial Least-Squares-Based Dimension Reduction (PLS).....	41
2.6.3 Factor Analysis (FA) .....	42
2.6.4 Linear Discriminant Analysis (LDA). .....	43
2.7 Graph Definition and origin of Graph Theory .....	44
2.7.1 Types of graphs.....	45
2.7.2 Graph connection.....	46
2.7.3 Network topologies.....	46
2.8 Graph-based feature selection methods.....	51
2.8.1 Graph-based filtering metrics .....	54



2.9 Graph clustering techniques .....	63
2.9.1 Partitioning clustering.....	63
2.9.2 Hierarchical clustering .....	64
2.9.3 Density-based clustering.....	64
2.9.4 Spectral clustering.....	65
2.9.5 Affinity propagation .....	65
2.9.6 Projective clustering .....	66
2.10 Graph Similarity measures .....	67
2.10.1 Pearson's correlation coefficient (PCC).....	67
2.10.2 Mutual Information.....	69
2.10.3 Spearman's rank correlation coefficient .....	70
2.11 Data discretization.....	71
2.11.1 Unsupervised discretization.....	72
2.11.2 Supervised discretization of gene expression data (GED).....	73
2.12 Machine learning.....	77
2.12.1 Supervised learning.....	78
2.12.2 Unsupervised .....	86
2.13 Research gap.....	93

<b>CHAPTER THREE .....</b>	<b>95</b>
<b>METHODOLOGY.....</b>	<b>95</b>
3.1 Study design .....	95
3.2 Data type and data source.....	96
3.3 Graph-based feature selection model algorithm.....	97
3.3.1 Checking the quality .....	98
3.3.2 Removing low quality reads .....	99
3.3.3 Experiment 4: Indexing and mapping to the reference genome .....	101
3.3.4 Experiment 5: Counting of the features .....	101
3.3.5 Experiment 6: Normalization.....	102
3.3.6 Experiment 8: Network construction.....	102
3.3.7 Experiment 10: Discretization .....	104
3.3.8 Experiment 11: Association rule mining .....	105
3.4 Experiment 12: Classification .....	109
3.4.1 Class balancing .....	110
3.4.2 Building classifier models .....	111
<b>CHAPTER FOUR.....</b>	<b>115</b>
<b>RESULTS AND DISCUSSION .....</b>	<b>115</b>

4.1 Data preprocessing results .....	115
4.1.1 Trimming low quality reads results .....	115
4.1.2 Indexing and mapping .....	117
4.1.3 Feature counting results .....	120
4.1.4 Normalization and differential expression analysis results .....	121
4.2 Graph construction .....	121
4.2.1 Graph threshold and module detection results.....	121
4.3 Discretization results .....	126
4.4 Apriori Algorithm-Based Association rule analysis.....	128
4.5 Model validation results .....	130
4.6 Association Rule Mining.....	133
4.7 Classification as an alternative feature selection method.....	141
<b>CHAPTER FIVE.....</b>	<b>146</b>
<b>CONCLUSIONS AND RECOMMENDATION FOR FUTURE RESEARCH.....</b>	<b>146</b>
5.1 Conclusions .....	146
5.3 Recommendations for future work.....	147
<b>REFERENCES.....</b>	<b>148</b>
<b>Appendix I: Author’s Publications during PhD study .....</b>	<b>194</b>



## LIST OF TABLES

<b>Table 3.1:</b> summary of datasets.....	97
<b>Table 3.2:</b> Error probabilities for assigning quality scores (Ewing & Green, 1998) .....	99
<b>Table 4.1:</b> Network topology for the top genes with a node degree greater than 8. ....	125
<b>Table 4.2:</b> Association rules among genes that showed significant upregulation after exposure to an attractant ( $\epsilon$ -nonalactone). .....	129
<b>Table 4.3:</b> Output from normalization and feature selection using PCA, RFE and graph-based approaches. ....	131
<b>Table 4.4:</b> Rules generated using Apriori from features selected using three different	134
<b>Table 4.5:</b> A summary of top ten rules generated from the two datasets after graph-based feature selection.....	140
<b>Table 4.6:</b> Classification results after feature selection. ....	142

## LIST OF FIGURES

<b>Figure 2.1:</b> a) RNA-seq data format; b) example of quality score encoding .....	11
<b>Figure 2.2:</b> Summary of the key steps in feature extraction from RNAseq data .....	12
<b>Figure 2.3:</b> Data structure of the hash table .....	17
<b>Figure 2.4:</b> Querying a hash table. ....	18
<b>Figure 2.5:</b> BWT algorithm applied to string 'AGGCT' .....	19
<b>Figure 2.6:</b> The characters (above) and its corresponding suffix array .....	20
<b>Figure 2.7:</b> (a)Undirected graph; (b) Directed graph; (c) Undirected graph (disconnected) (d) complete undirected, graph.....	46
<b>Figure 2.8:</b> Small-world network.....	48
<b>Figure 2.9:</b> Scale free graphs.....	49
<b>Figure 2.10:</b> disease association module.....	53
<b>Figure 3.1:</b> Workflow of the study.....	95
<b>Figure 3.2:</b> Graph-based model for feature selection and phenotype prediction. ....	98
<b>Figure 3.3:</b> Discretization process.....	105
<b>Figure 3.4:</b> Apriori algorithm.....	106
<b>Figure 4.1:</b> Comparison of the output from three trimming algorithms. ....	116
<b>Figure 4.2:</b> Comparison of the various indexing and mapping algorithms.....	118

<b>Figure 4.3:</b> Final feature counts after feature extraction.....	120
<b>Figure 4.4:</b> a) Scale-free fit index versus soft-thresholding power, b) Grouping of features into modules based on the expression patterns, c) Global network for all features, d) Global network statistics .....	122
<b>Fig. 4.5:</b> a) Co-expression networks.....	123
4.5c) Network summary statistics before filtering.....	124
<b>Figure 4.6:</b> A sample output of the discretization process.....	127
<b>Figure 4.7:</b> Features selected by each of the methods from the two datasets .....	131
<b>Figure 4.8:</b> Network diagrams for the 2 datasets .....	133
<b>Figure 4.9a:</b> Summary of rules with lift and support MCC.....	136
<b>Figure 4.9b:</b> Rules maximum support and lift after filtering using Edge Percolated Component .....	137
<b>Figure 4.9c:</b> Rules maximum support after filtering using ECC .....	138
<b>Figure 4.9d:</b> Rules maximum support after filtering using Degree .....	139
<b>Figure 4.10:</b> comparison of classifiers accuracy before and after feature selection for dataset GSE60052 .....	143
<b>Figure 4.11:</b> comparison of classifiers accuracy before and after feature selection for dataset GSE81089 .....	144

## LIST OF APPENDICES

<b>Appendix I:</b> Author's Publications during PhD Study .....	194
---	-----



## ABBREVIATIONS AND ACRONYMS

<b>ANN</b>	Artificial Neural Network
<b>ARM</b>	Association Rule Mining
<b>BWT</b>	Burrows-Wheeler Transform
<b>CNN</b>	Convolutional Neural Network
<b>CS</b>	Chi-Square
<b>CST</b>	Chi-Square Tests
<b>DNA</b>	Deoxyribonucleic Acid
<b>DT</b>	Decision Tree
<b>EFD</b>	Equal Frequency Discretization
<b>FA</b>	Factor Analysis
<b>FE</b>	Feature Extraction
<b>FN</b>	False Negative
<b>FP</b>	False Positive
<b>FS</b>	Feature Selection
<b>GDA</b>	Gaussian Discriminative Analysis
<b>GED</b>	Gene Expression Data
<b>ICA</b>	Independent Component Analysis

<b>IG</b>	Gain Ratio
<b>IT</b>	Information Technology
<b>KNN</b>	K-nearest Neighbors
<b>KS</b>	Kappa Statistic
<b>LCP</b>	Longest Common Prefix
<b>LDA</b>	Linear Discriminant Analysis
<b>LR</b>	Logistic Regression
<b>LWLR</b>	Locally Weighted Linear Regression
<b>MAE</b>	Mean Absolute Error
<b>MII</b>	Multidimensional Interaction Information
<b>ML</b>	Machine Learning
<b>MLP</b>	Multi-Layer Perceptron
<b>MMP</b>	Maximum Mappable Prefixes
<b>MRMR</b>	Minimum Redundancy Maximum Relevance
<b>NB</b>	Naïve Bayes
<b>NCA</b>	Neighborhood Component Analysis
<b>NGS</b>	Next Generation Sequencing
<b>NSCLC</b>	Non-Small Cell Lung Cancer

<b>PCA</b>	Principal Component Analysis
<b>PCC</b>	Pearson's Correlation Coefficient
<b>PLS</b>	Partial Least Squares
<b>RF</b>	Random Forest
<b>RF</b>	Relief-F
<b>RFE</b>	Recursive Feature Elimination
<b>RMSE</b>	Root Mean Squared Error
<b>RWR</b>	Random Walk with Restart
<b>SA</b>	Suffix Array
<b>SCLC</b>	Small Cell Lung Cancer
<b>SMO</b>	Sequential Minimal Optimization
<b>SMOTE</b>	Synthetic Minority Oversampling Technique
<b>SVM</b>	Support Vector Machine
<b>TN</b>	True Negative
<b>TP</b>	True Positive
<b>WCSS</b>	Within Cluster Sum of Squares

## ABSTRACT

High throughput sequencing technologies generate large volumes of data and this effectively ushers' life sciences into the big data realm. Data generated using these technologies is oftentimes noisy or high-dimensional and therefore several preprocessing steps for its computational analysis are required. Dimensionality reduction methods focus on evaluating each feature individually instead of putting into consideration the interactions or dependencies between features. These relationships are very important because they reflect the functional/ phenotypic aspect in living systems. The aim of this study was to develop a graph-based network feature selection model for gene-phenotype prediction in high dimensional RNAseq data. Three different datasets (RNAseq data from; antennae of *Glossina morsitans morsitans*, Small Cell Lung Cancer (SCLC) and Non-small Cell Lung Cancer (NSCLC)) were used. Pre-processing involved quality checking, adapter trimming, contamination removal and quality filtering. Differential expression analysis was done, and genes were considered differentially expressed and retained for further analysis if the test statistics p-value (adjusted for false detection rate) (FDR) was less than 0.05. Feature selection was performed using Principal Component Analysis (PCA), Recursive Feature Elimination (RFE) and a Graph-based approach. Equal Frequency Discretization (EFD) was used to transform the selected features from a continuous or numerical attributes into discrete values. Association rules were generated using a minimum support value between of 0.5 and 0.9, minimum confidence value of 0.9 and lift of  $\geq 2$ . Features from the three feature selection techniques were classified using three classifiers namely Naïve Bayes, Sequential Minimal Optimization (SMO) and Multilayer Perceptron. Results from the quality trimming showed that the window-based algorithm performed better than the other two approaches whereby the percentage of the surviving reads ranged between 83.39% and 90.87%. Mapping results showed that Burrows wheeler algorithm performed better than Bowtie2 in terms of the alignment across all the samples with accuracy values between 93% and 97.97%. During differential gene (feature) expression analysis, 2,097 low-count features were filtered out leaving a final tally of 10,921 features. Three global networks with 2,110 nodes and 4,783 edges, 990 nodes and 3154 edges and 876 nodes and 3676 edges were generated from three datasets used in this study. The resulting networks were further filtered, and the final reduced networks had 51 nodes and 148 edges, 134 nodes and 396 edges, and 81 nodes and 169 edges respectively. The proposed graph-based feature-selection approach provided 15 and 36 non-redundant rules, respectively, from the two datasets at a support of 0.5 confidence value of 0.9 and a lift of 2. PCA and RFE feature-selection methods did not generate any rules at a support of 0.5. The lower support values provided by RFE feature selection approach implies that the features selected by this method were negatively correlated. For the PCA-based feature selection, support ranged between 0.405 and 0.425 which was lower than the support of the rules generated by the graph-based feature selection approach. The results of classification before and after feature selection showed a reduction in classifier model building time with minimal effect on accuracy. This study demonstrates that graph-based feature selection approach combined with association rule mining can be very useful in associating genes with a known function

with those with unknown function for phenotype prediction based on gene expression levels.

## CHAPTER ONE

### INTRODUCTION

#### 1.1 Background of the study

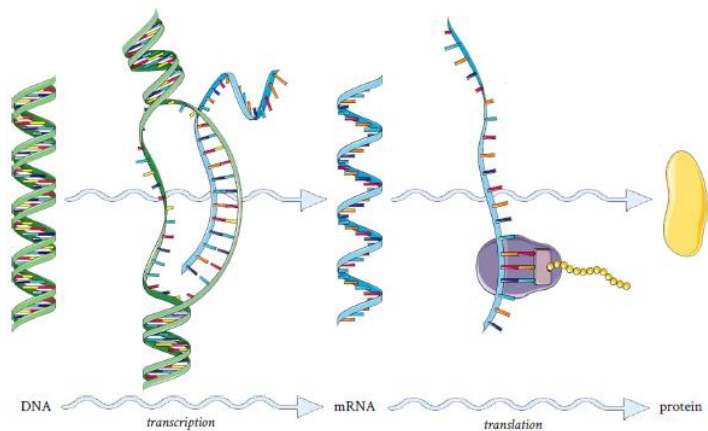
Microarray and Next Generation Sequencing (NGS) are high-throughput technologies that continuously generate large volumes of high dimensional biological data (Ai *et al.*, 2018). These advances in high throughput sequencing and digitalization has effectively ushered life sciences into the realm of big data. Discovery of meaningful associations in this kind of data consumes a lot of time and is computationally demanding (Curtin *et al.*, 2015). Biological data is characterized by high volume, velocity, and variety (Gärtner & Hiebl, 2017). This type of data is highly heterogeneous due to inherent biological principles and experimental designs. In biological systems, functional relationships exist between genes, proteins, and pathways. Therefore, big data analytics has over the years become an indispensable tool for managing bioinformatics data.

##### 1.1.1 Genes and gene expression

Gene expression is the measurement of the activity (the expression) of thousands of genes at once, to create a global picture of cellular function. These profiles can, for example, distinguish between cells that are actively dividing, or show how the cells react to a particular treatment. Many experiments of this sort measure an entire genome simultaneously, that is, every gene present in a cell.

Deoxyribonucleic acid (DNA) is an organic molecule found in all living cells and it carries the genetic instructions required for cellular function. It is a long sequence of a combination of the four nucleotides A, C, G and T (which are named after their respective bases adenine, cytosine, guanine and thymine Watson & Crick, (1953). A gene is a specific region within this DNA sequence (Alberts *et al.*, 2013) and it encodes for a protein. To synthesize a particular protein, the information on the DNA is transformed to a messenger ribonucleic acid (mRNA) in a process referred to as transcription. After some

modifications the mature mRNA will be translated to the final protein product which is a protein with a particular function. Some of the gene products are not proteins but functional ribosomal RNAs. In living systems, these processes are well coordinated and regulated to meet the cellular needs. The whole process from a gene to a functional gene product is termed gene expression.



**Figure 1.1 Central dogma of molecular biology:** DNA is transcribed to RNA; RNA is translated to protein. (Les Laboratoires Servier, 2018)

### 1.1.2 Gene phenotype

Phenotype is the physical characteristic of an organism which is reflected as appearance, behavior, or development. Genotype which is the set of genes carried by the organism as well as influence by the environment which has an influence on genes determines the organism's phenotype. The impact of a gene on an individual's phenotype therefore is dependent on other genes or gene products (Weighill *et al.*, 2019)

### 1.1.3 Sources of big data

Big data is derived from a variety of sources, including spreadsheets, traditional databases, text documents, and digital data streams. Internet for example provides data such as navigation and search history from different browsers and from social networks. On the

other hand, mobile devices provide ubiquity that enable collection of real-life behaviors using embedded sensors such as GPS, cameras etc. Social media platforms such as Facebook, Twitter and LinkedIn are other sources of big data with tremendous data and unprecedented opportunity for big data analytics. (Huang *et al.*, 2015).

#### **1.1.4 Curse of dimensionality**

Big data is associated with an increase in dimensionality which eventually leads to exponential growth of volume of data required for meaningful analysis (Conesa *et al.*, 2016, Nia *et al.*, 2020). This phenomenon was earlier defined as a curse of dimensionality. Dimensionality describes the total number of features or attributes that are present in a dataset (Khare *et al.*, 2019). The problem is associated with the increase in dimensions or characteristics  $p$  that describe every record  $[n]$  in the database. Analysis of high dimensional data, with more features ( $p$ ) than observations ( $N$ ) ( $p > N$ ), places significant computing costs and memory computational usage attributes (Rathor & Gyanchandani, 2017). When the total number of features used increases, the accuracy and the performance of machine learning algorithms decreases (AlSumairi *et al.*, 2020). The effect of high dimensional data on training set is a decrease in algorithm performance when the dimensionality increases (AlSumairi *et al.*, 2020). Therefore, data mining and machine learning are strategies big data analytics to manage growing volumes of data, especially in bioinformatics (Curtin *et al.*, 2015).

#### **1.1.5 Feature engineering**

Feature engineering is the process of establishing features by use of domain knowledge thus improving machine learning performance. Selection of the most relevant features is regarded as one of the most time-consuming preprocessing tasks in machine learning (Najafabadi *et al.*, 2015). As the data size increases, difficulties associated with feature engineering arise (Najafabadi *et al.*, 2015). Many learning algorithms assume that data to be processed even big data can be stored in memory or in an entirely single file on a disk (Kumar *et al.*, 2013). However, if the size of data compromises this principle, all



algorithms in that family are affected. This challenge is known as the curse of modularity (Rathor & Gyanchandani, 2017). It is also assumed that algorithms used in machine learning learn better with a bigger volume of data and provide better or accurate results (Grolinger *et al.*, 2014). However, massive volume of datasets imposes several challenges since many machine learning algorithms were designed to handle small datasets, with the assumptions that entire datasets can fit or be stored in memory (Sharma *et al.*, 2019). The second assumption is that the whole or entire dataset could be availed for processing during the model training phase. However, big data does not adhere to these assumptions, something that makes traditional algorithms unfeasible or significantly impedes their performance in terms of execution time (Arora, 2019).

### **1.1.6 Processing performance**

Computational analysis of big data is always challenging due to computational complexity. Consequently, an increase in scale leads to unimportant operations becoming very costly. A good example is the support vector machine (SVM) algorithm's which has a training time complexity of  $O(m^3)$  with a space complexity of  $O(m^2)$ , where  $m$  implies the number of available samples for training (L'heureux *et al.*, 2017). Therefore, increase in the size  $m$  leads to extreme effects on the training time and memory that is needed by the SVM algorithm. This is not computationally feasible when handling very huge datasets. Other machine learning algorithms that have exhibited increase in time complexity are Principal Component Analysis (PCA), Logistic Regression (LR), Gaussian Discriminative Analysis (GDA) as well as locally weighted linear regression (LWLR). All these have a time complexity of  $O(mn^2 + n^3)$ , with  $m$  being the total number of samples and  $n$  representing the total number of features (Hu *et al.*, 2020). In all the above-mentioned algorithms, computational time increases exponentially when data size is increased rendering the algorithms unfeasible for very large datasets (Arora, 2019). Another challenge posed by biological data is class imbalance, a term used to describe uneven sample number or biased distribution across the classes. The distribution ranges

from a little skew to a very severe imbalance, where there are fewer samples in the minority class as compared to hundreds in the majority class (Wei & Sekiya, 2021).

### **1.1.7 Dimensionality reduction.**

Dimensionality reduction is the process of transforming high dimensional representation of data to low dimensional representations without losing any important information (Jindal & Kumar, 2017; Nguyen & Holmes, 2019). Using this approach, a lower and reduced dimensional feature space is mapped onto a higher dimensional feature space thereby developing a linear separability (Hira & Gillies, 2015). The massive growth in high dimensional data has led to development of various dimensionality reduction techniques (Zebari *et al.*, 2020). These approaches are supportive and essential due to their ability to map, trim, distinguish and exemplify datasets through conversion from high dimensional space to a much lower dimensional space by influencing the significant variables (Arowolo *et al.*, 2017). The benefits of dimensionality reduction include reduced data storage space, less computation time, removal of irrelevant, and noisy data etc. Another benefit is the ability to examine patterns more clearly as well as improved classification accuracy (Zebari *et al.*, 2020). Some algorithms produce good results or improved performance when there are fewer number of dimensions. Therefore, these benefits make machine learning experts spend most of their time on data cleaning phase and feature engineering (Zheng & Casari, 2018).

There are two major approaches of addressing the challenges associated with high dimensionality in data. These techniques are categorized into feature selection and feature extraction. Feature selection techniques discover only the relevant features from the original dataset using objective measures (Arowolo *et al.*, 2021). Feature extraction is achieved by filtering out all the irrelevant, redundant as well as noise that is present in high-dimensional dataset. The major feature selection approaches have been classified as Filters, Wrappers and Hybrid/ embedded feature (Jindal & Kumar, 2017). Filter methods use relevant model learning algorithms that are independent of any classifier and pick only the relevant features described by earlier study by Kumar & Minz, (2014). They rely on

the uniqueness of the data provided. Statistical procedures are used to calculate feature scores since they are robust against the problem of over-fitting as compared to other feature selection techniques and procedures (Manikandan & Abirami, 2021). The major drawback of filter-based approaches is that they ignore classification interaction as well as the interdependencies amongst the features. This may lead to the most relevant features not being picked (Mafarja & Mirjalili, 2018).

Wrapper-based feature selection methods are based on specific machine learning algorithm that is used in picking the relevant features while considering the learning algorithm that will be used. It evaluates the quality of the selected features using a precise classifier which runs several times to assess the quality of features based on the accuracy of the assigned scoring model (Ray *et al.*, 2021). A wrapper-based feature selection method also performs optimal feature selection by calculating estimated accuracy for every feature using induction algorithm (Aziz *et al.*, 2018). The major advantage of using wrappers as compared to filter techniques is that they locate the most constructive features and optimize selection of features that are required for the learning algorithm (Kumar & Minz, 2014). Wrapper processes have high computational complexity because a feature subset is chosen and the classifier is run on it in each iteration, followed by the computation of classification accuracy using the resultant confusion matrix. This process makes it require more computational resources because the algorithms used execute iteratively (Hammami *et al.*, 2019).

Embedded feature selection methods are generally guided by a learning process which is called nested subset method (Eswari *et al.*, 2015). They measure the relevance of feature subsets, and the entire feature selection is done as a training process while optimizing the learning algorithm's performance. This enables the usage of available data to generate faster solutions. The benefits of both filter and wrapper methods are combined in embedded approaches and are dependent on the machine learning algorithm used (Maldonado & López, 2018). Embedded methods have lower computational requirements, less prone to over-fitting and they provide better classifiers by considering the feature dependencies thus providing faster solutions. Their major drawback is that they

take dependent classification decisions, and this affects the selected features due to the varying hypothesis of different classifiers (Abdulrazzaq & Saeed, 2019).

The second approach in dimensionality reduction is feature extraction which takes the most important features from a dataset and express them in a lower-dimensional space. The new features are merged into a linear or nonlinear combination of the original features. In this case, dimensionality reduction can be done in combination with other machine learning algorithms to enhance the model's accuracy and other parameters. Appropriate dimensionality reduction algorithms can be evaluated in terms of improvement in performance metrics such as accuracy, sensitivity, specificity, recall, robustness, computational scalability, and computational cost etc. (Sun *et al.*,2019). However, in the process of mapping from a high-dimensional space to a low-dimensional space, feature extraction approaches suffer from erroneous outputs, resulting in a loss of data interpretability (Malekipirbazari *et al.*, 2021).

To overcome the challenges associated with the feature selection and feature extraction techniques from high throughput sequencing data, this study provides a graph-based feature selection approach that takes into account the inherent interactions between features. Three publicly available RNASeq datasets were used. Various data mining tools were applied to reduce the dimensionality of the big data. Informative features were then extracted using a graph-based method. Thereafter association rule mining was used to predict the potential phenotype of unknown features.

## **1.2 Problem statement**

Advances in high throughput sequencing and digitalization of almost all procedures has effectively ushered life sciences into the realm of big data (Ai *et al.*, 2018). Biological data is regarded as high dimensional data because it is characterized by more features than observations (Rathor & Gyanchandani, 2017). Discovery of meaningful associations in this kind of data is a very challenging task in bioinformatics (Drouin *et al.*, 2019). Challenges arise because current approaches fail to consider the relationships between

selected/extracted features during feature selection process. These approaches focus on evaluating each feature individually while ignoring interactions or dependencies between features. In living systems, relationships between features are very important because they determine the function/ phenotype. Therefore, the selected features mostly have no direct relationship that can be used to associate those with known phenotype with those with unknown phenotype. This makes the prediction of the possible phenotype almost impossible. Therefore, there is a need for alternative feature selection approaches picking only related features/genes for predicting the phenotype of unknown features/genes based on their association with those with an assigned function.

### **1.3 Main objective**

The main objective of this study was to develop a graph-based feature selection model in association with association rule mining for genes' phenotype prediction

### **1.4 Specific objectives**

1. To analyze techniques for feature extraction and selection in high dimensional RNAseq data.
2. To develop a graph-based feature selection model for phenotype prediction
3. To determine association patterns between the selected features by graph-based method for phenotype prediction.
4. To validate the graph-based feature selection model

### **1.5 Justification**

High throughput sequencing technologies generate large volumes of data that can be useful in addressing important biological question. Different feature selection and extraction approaches have been used in dimensionality reduction to deduce meaningful information from big data. However, these methods don't consider the inherent relationship amongst the selected/extracted features, which is a key characteristic of biological systems. A graph or network provides a representation of related features

whereby nodes and edges represent features and the relationship amongst features respectively. In biological data a graph would be a more suitable method for visualizing and interpreting physical interaction, reaction, regulation, and correlation between the features once the dimensionality has been reduced. Association rule mining can then be used to predict the phenotype of novel features using the concept of market basket analysis. This study combined the above-mentioned approaches to provide a graph-based phenotype prediction model.

### **1.6 Scope of the study**

This thesis explores techniques applied in feature extraction and feature selection for biological data analysis. Since biological features tend to associate in a certain way, graph theory was used to cluster features using the guilt by association principle. Machine learning and association rule mining were used to generate rules and identify meaningful associations between various features (genes).

### **1.7 Knowledge Contributions**

In this study a graph-based feature selection model for genes phenotype prediction was proposed. On feature extraction evaluation part, this study contributed by evaluation of the most optimal techniques for converting RNASeq data to continuous values. A graph-based feature selection model was provided that can be used for analysis of similar type of data. A detailed comparison with other popularly used feature selection techniques on high dimensional data was also done in this study. The concept of market basket analysis was the key contribution of this study in prediction of the possible function of genes based on how genes/items appeared frequently in the same transaction.

Based on the results of this study, it is evident that several critical preprocessing steps are required before feature selection and feature extraction can be objectively used to make predictions based on next generation sequencing data. This would lead to faster discovery

of biomarkers for rapid health screening and diagnostics especially for cancer and other metabolic diseases.

## **1.7 Thesis organization**

This thesis is divided into five chapters organized as follows:

- Chapter 1 Introduces the study by giving an overview of the study, describing the problem statement, research objectives, justification, and thesis organization. The next four chapters are organized as follows.
- Chapter 2 presents the literature review and starts with general introduction of bioinformatics theoretical background and of different data structures used in high dimensional biological data as well as mapping and feature counting. A description of the feature selection and extraction methods used in high dimensional data and related literature on the same is discussed. analysis of different machine learning and their application in high dimensional biological data and the methods of model evaluation follows. Discretization methods and their theoretical basis are also described followed by association rule mining.
- Chapter 3 presents the methodology used in this study, data type and data source, data preprocessing and the experiments used to achieve the study objectives.
- Chapter 4 presents the output or results of the experiments carries out in chapter three together with the discussion of the results in relation to the previous findings presented inform of tables, figures, and graphs.
- Chapter 5 provides the conclusions, knowledge contribution and recommendations for further studies

## CHAPTER TWO

### LITERATURE REVIEW

This chapter introduces bioinformatics data format, data structures that are used in storage of next generation sequencing data, feature selection and extraction techniques and their working principles. Graph theory and how it has been used in dimensionality reduction followed by data discretization are also discussed. Related literature on machine learning in big data analysis is thereafter presented followed by the research gap.

#### 2.1 Introduction to bioinformatics

Data generated by sequencing technologies such as Illumina have a certain format as shown in Figure 2.1. Every single record also called sequence read has four lines as indicated below:

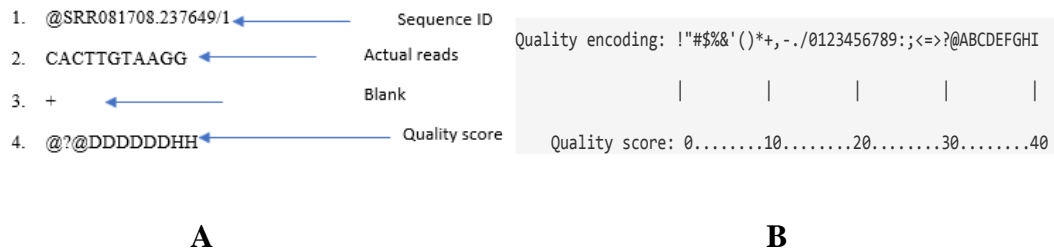


Figure 2.1: a) RNA-seq data format; b) example of quality score encoding

As shown in figure (2.1 a), the first line always starts with '@' which is followed by information about the read, the second line is the actual DNA sequence, the third line always starts with a '+' sign and sometimes contains similar information as in line 1 and other times acts as a place holder with no information. It depends on the sequencing technology used. The fourth line contains a string of characters which represents the quality scores and must contain the same number of characters indicated in line 2. The characters in the sequence are encoded with quality scores of ASCII as shown in (Figure 2.1b) shows an example of quality scores mapping where each base quality is an ASCII



encoded (Figure 2.1b). Once the data has been preprocessed, reads that pass the quality score are mapped to a reference genome followed by counting the mapped reads as summarized in Figure 2.2.

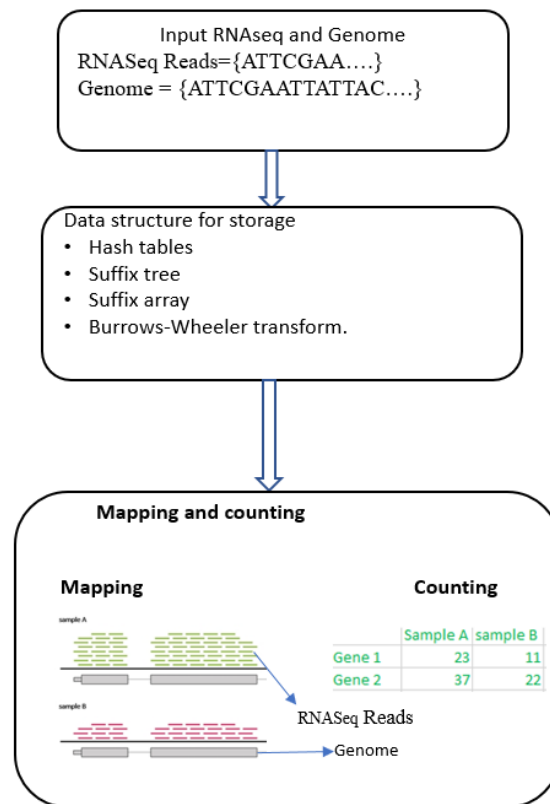


Figure 2.2: Summary of the key steps in feature extraction from RNAseq data

## 2.2 Data structures used in bioinformatics

Algorithms and data structures have been regarded as fundamental concepts in computer science, and therefore any understanding of bioinformatics that extends beyond the basic usage of common tools and methodologies necessitates at least a basic grasp of these concepts. Many prominent methods for interpreting sequencing data rely on string matching, with most ways focused on first recognizing short, fixed length read substrings. These are referred to as k-mers, with k representing the substring length. While these k-

Mer algorithms are quite distinct, storing and querying a set of k-mers has emerged as a common underpinning component. Because of the enormous scale of these datasets, reducing their storage requirements and query times has become a separate field of study (Chikhi *et al.*, 2021). Next-generation sequencing (NGS) is a technology which is used to determine the order of nucleotides order in an entire genome or specific regions of DNA or RNA. NGS has a broad spectrum of applications in cancer genomics, however, the bioinformatic analysis which is involved in the transformation of the raw “ATGC” sequence to meaningful genomic information such as gene expression abundance or gene mutations is a non-trivial work. This technology has transformed the field of life sciences by enabling fast and cost-efficient generation of big volumes of data. However, NGS is a computationally intensive process that requires auxiliary data structures (Wu *et al.*, 2016). Basic bricks in bioinformatics begin by first constructing a data structure that is called an index. Most of these have been built using data structures that are simple and basic such hash tables, suffix trees, suffix arrays etc. Therefore, an index supports fast queries while using reasonable amount of memory (Salikhov, 2017). Despite the fact that various data structures are created differently and support distinct sorts of operations, they are all utilized to solve the same challenge in bioinformatics (Salikhov, 2017).

### **2.2.1 Hash tables**

A hash table is a type of data structure that stores collections of associative arrays with  $(key, value)$  pairs and allows three operations which are:  $insert(key, value)$ ,  $find(key)$  and  $remove(key)$ . For a successful mapping of keys to integers starting from 0 to  $m - 1$ , a hash function  $h$  is used by the hash table data structure which is a representation of the indexes of an array  $A$  of size  $m$ . If a key  $k$   $h(k) = i$ , then element  $(k, v)$  is put in a slot  $i$  of the array. Sometimes different keys can be mapped to the same index, and this can lead to a situation called a *collision*. A collision is a major drawback of hash table data structures but there are two ways of dealing with it (Wang *et al.*, 2021). The first method called separate chaining works by storing a list of elements that are contained in every slot of array  $A$ , and if a need arises to add an extra element with key  $k$  in slot  $i$ , it’s just

appended it to the end of its corresponding list. The second strategy is known as open addressing whereby the algorithm searches for an empty slot for insertion of key  $k$  if the corresponding position is not empty. Therefore, hash tables offer very fast operations. On average, they all work in the same amount of time, depending on the hash table and hash function parameters. One operation in a hash table can take  $O(n)$  time in the worst-case scenario, with  $n$  being the number of elements that have been inserted. Practically, hash tables are faster than other data structures because they store key: value pairs which allows a search to be done using a key. The choice of the appropriate hash function is very crucial to obtain best performance of a hash table data structure. When the choice of hash function is good, then there is an insertion of  $n$  elements into array of size  $k$ . This makes an average search of a single element to work in  $O(1 + \frac{n}{k})$  time when a separate chaining strategy is selected for addressing the challenge of collisions. In this case  $\frac{n}{k}$  is a load factor used to show the number of elements in terms of average that have been inserted in the same slot of an array. Even though hash tables outperform many other data structures in terms of query speed, they are typically quite memory intensive (Petrillo *et al.*, 2019).

### 2.2.2 Suffix tree

A compressed *trie* of all suffixes in an input string is known as a suffix tree. One of the most researched data structures in stringology is this type of data structure, which is used in text string processing. (Gog *et al.*, 2014). A suffix tree is usually constructed in linear time and space so that it can take time  $O(|P|)$  to search for a pattern  $p$  and time  $O(|P| + occ)$  to report all the occurrences of  $P$ . Since it is usually a pointer-based structure, then the suffix trees require  $O(n)$  words in a computer but the bits of memory required is of  $O(n \log(n))$ . This data structure is memory intensive making it unpopular in real program implementation. MUMmer is an acronym from "Maximal Unique Matches", works by locating maximal unique matches between two sequences using a suffix tree data structure and therefore it requires a memory higher than 45 GB to run for analysis of the human genome (Brinda, 2016).

### 2.2.3 Suffix array

A suffix array is a representation of lexicographically sorted suffixes of an array of a string which has been used mainly in string processing applications (Wu *et al.*, 2019). Compared to suffix tree, a suffix array is more compact and simpler. Construction of a suffix array is done in linear time using a lexicographic traversal of the suffix tree. Suffix array works by using the principle of having permutations of the positions of  $T$  which is picked in a lexicographical order of the corresponding suffixes. To search for a pattern of  $P$  all the suffixes of string  $T$  that has the prefix  $P$  are identified. These suffixes are then sorted in a lexicographical order which makes the suffixes which are prefixed by  $P$  such that they are in a consecutive order in the suffix array that forms an interval.  $LP$  (left prefix) and  $RP$  (right prefix) define the borders, which are found using a binary search, that has a time complexity  $O(|P| \log(|T|))$ . To reduce the search time complexity to  $O(|P| + \log |T|)$ , longest common prefix (LCP) is usually provided. Unlike suffix tree, suffix array data structures are more preferred, but they still suffer the challenge of high memory consumption (Shrestha *et al.*, 2014). Earlier study by Abouelhoda *et al.*, (2004) demonstrated that any algorithm that uses the suffix tree as the data structure can be successfully substituted by an algorithm that can use a certain variant of suffix arrays and it use the same time complexity to solve the same problem. Therefore, suffix arrays have the capability of fully replacing suffix trees in any practical application.

### 2.2.4 Burrows-Wheeler transform

The Burrows-Wheeler transform (BWT) is a textual transformation data structure that is frequently used for compression and indexing and is specifically referred to as BWT-index. Due to the reversibility of BWT and the accompanying features of the generated strings as identical letters, it has been used widely as data structure for storage of sequence characters (Brinda, 2016). BWT works by appending a special character (Sigil) '\$' to the end of the string to complete the string transformation, and all of its cyclic shifts are sorted in lexicographical order. The very final column of the matrix, commonly known as the Burrows-Wheeler matrix, has these shifts in its rows, resulting in the BWT. A BWT is

easily extracted from the suffix array that has been constructed using other data structures algorithms discussed above. Earlier studies by Rosone & Sciortino, (2013) and Kucherov *et al.*, (2013) studied the aspects of BWT that included its relations to combinatorics on words and its statistical properties. To obtain the resulting interval in  $|P|$  steps, the entire interval of the suffix array corresponds to an empty pattern, that is obtained by processing  $P$  from right to left (Brinda, 2016). After creating a data structure and indexing the genome based on the data structure, the next step is mapping the raw reads into the reference genome also based on the data structure created in the previous stage. The next section describes the mapping algorithms and their theoretical background/ working principle

## **2.3 Mapping algorithms**

### **2.3.1 Algorithms based on hash table**

An empty hash table consist of an array of empty buckets (empty boxes). As items are added into these empty buckets/boxes they usually become the lists. Also associated with this table is a hash function. The hash function maps each distinct key or each distinct 3 – *mer* unto one of the buckets in this array (Figure 2.3).

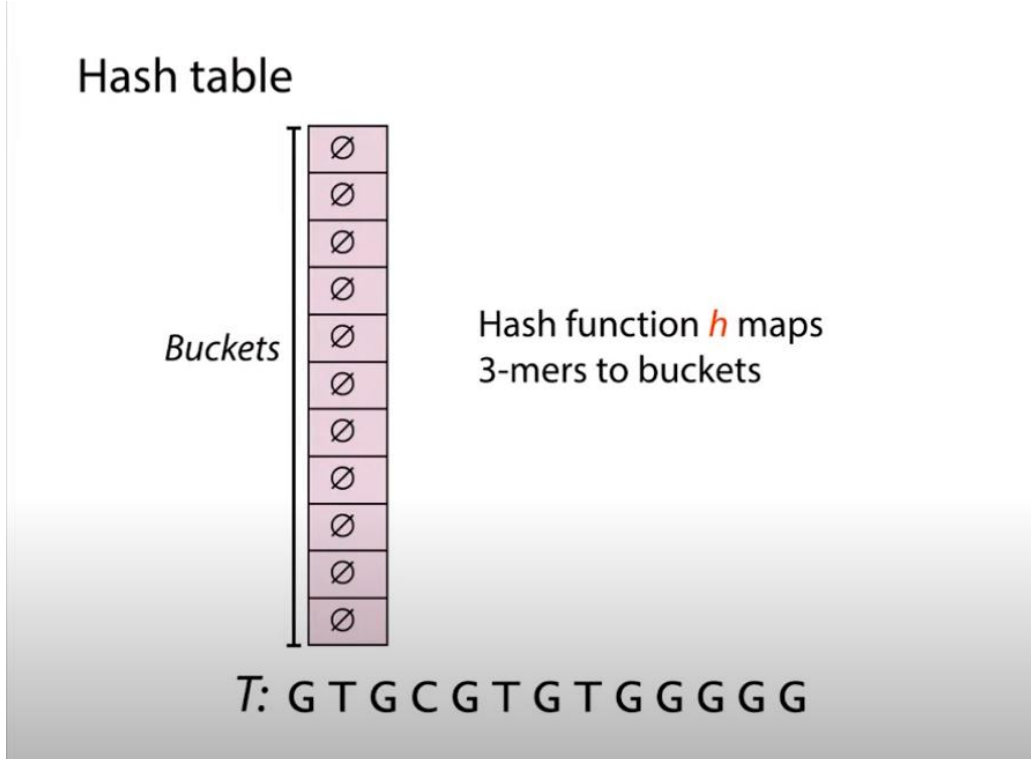


Figure 2.3: Data structure of the hash table

### Querying the hash table

The pattern  $p$  which is the sequence data is used to query the hash table. If GGG is picked so that the index can show all the offset where GGG occurs within the text  $T$  which in this case is the reference genome.

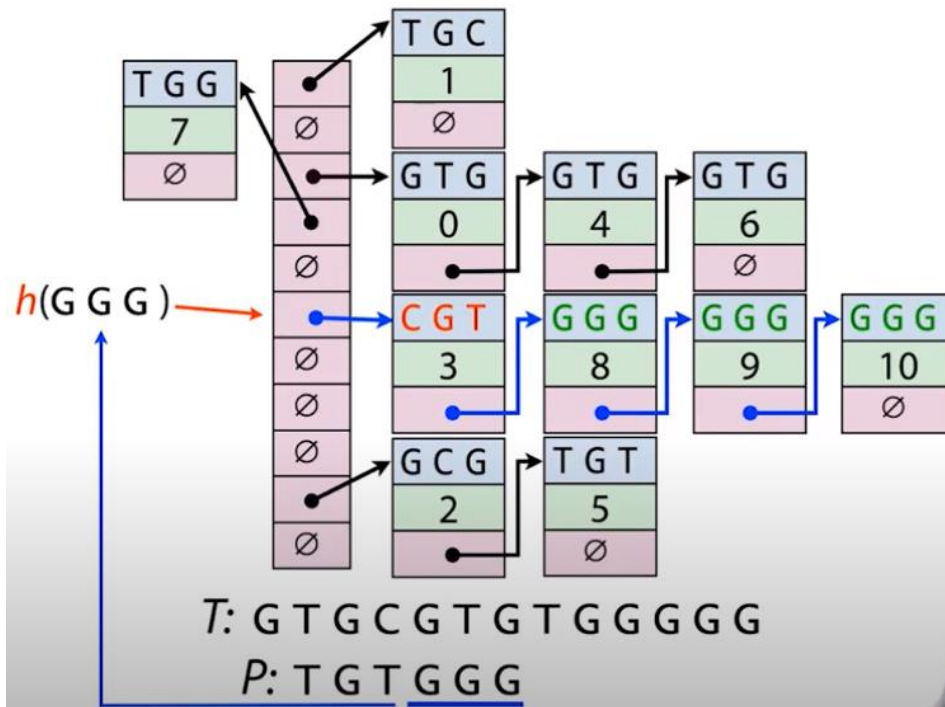


Figure 2.4: Querying a hash table.

First, the hash function is used to map the triple G to the bucket, the bucket with red pointer is the one looked at because it's the only one with triple G. The first bucket is ignored, and the next three cases is triple G. Therefore, the corresponding key value in that offset is what the index will report back as being the offset of *T* where triple G occurs and the index hits at 8,9 and 10 (Figure 2.4).

### 2.3.2 Algorithms based on Burrows-Wheeler transform

BWA is an alignment algorithm that employs a suffix array (SA). Indexes are assigned again according to their order throughout the process of producing the SA to specify new indexes using Burrow's wheeler matrix alignment. Because of the changed SA index, it is now possible to find places inside the data in string form. (Kim *et al.*, 2020) a demonstration of BWT is shown in figure 2.5.

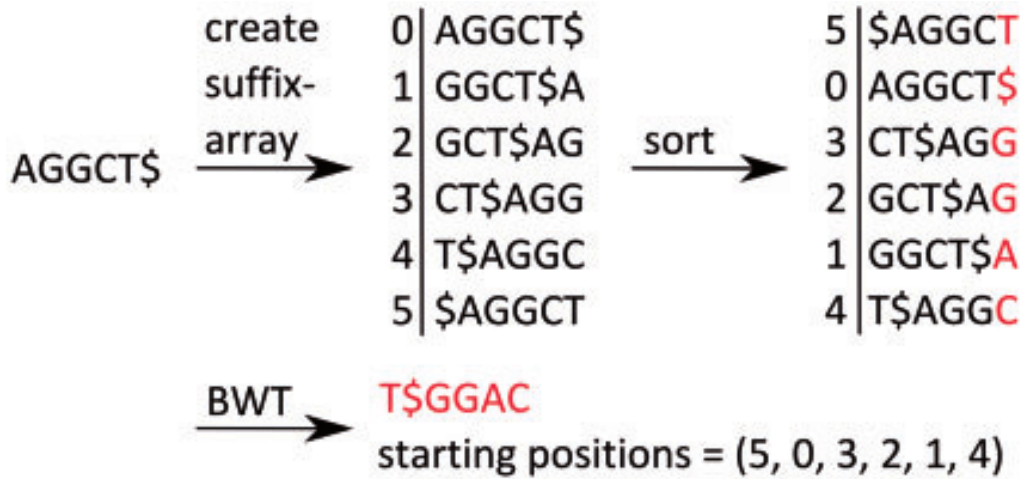


Figure 2.5: **BWT algorithm applied to string 'AGGCT'**

The BWT algorithm is applied on the string 'AGGCT\$,' where '\$' is the string's final character and is lexicographically smaller than all the other characters in the string. A suffix array is formed first, and then it is lexicographically sorted. Only the last column of characters and the order of the starting places are preserved in the original string (each of which has the same length as the string) are saved after the BWT. The memory needs are reduced to a linear scale in relation to the size of the string because of this character storage process (Kloetgen *et al.*, 2014).

### 2.3.3 Algorithm based on Suffix array

Algorithms that use Suffix Array (SA) searches an entire genome to find Maximum Mappable Prefixes (MMP). The algorithm searches for the junction position in the read sequence  $r_j$  that yields the maximum score by finding the maximum of the following quantity (Dobin *et al.*, 2013):



$$\left. \begin{array}{l} \max_{r_1 < r_j < r_2} \left\{ \sum_{r=1}^{r_j - r_1} \left[ \begin{array}{l} 1 \text{ if } R(r_1 + r) = G(g_1 + r) \text{ and } (r_1 + r) \neq G(g_1 + r + \Delta) \\ -1 \text{ if } R(r_1 + r) \neq G(g_1 + r) \text{ and } (r_1 + r) = G(g_1 + r + \Delta) \\ 0 \text{ otherwise} \end{array} \right] \right\} \\ P_{gap}^{(r_j)} \end{array} \right\} \quad (2.1)$$

Where R and G are reads (query) and genome sequences respectively, coordinates r1, r2, g1, g2.  $\Delta \equiv (g_2 - g_1) - (r_2 - r_1)$  is the alignment gap with the corresponding gap penalty  $P_{gap}^{(r_j)}$ . The amount of unmapped query sequence bases between the mapped seeds determines the algorithm's complexity, i.e.,  $r_2 - r_1 - 1$ . An example of suffix array is shown in figure 2.6 below:

0	1	2	3	4	5	6	7	8	9	10	11	12	13
t	g	t	g	t	g	t	g	c	a	c	c	g	\$

0	13	\$
1	9	accg\$
2	8	caccg\$
3	10	ccg\$
4	11	cg\$
5	12	g\$
6	7	gcaccg\$
7	5	gtgcaccg\$
8	3	gtgtgcaccg\$
9	1	gtgtgtgcaccg\$
10	6	tgcaccg\$
11	4	tgtgcaccg\$
12	2	tgtgtgcaccg\$
13	0	tgtgtgtgcaccg\$

Figure 2.6: **The characters (above) and its corresponding suffix array** (vertical) together with the matching suffixes on the right and the position index on the left (Shrestha *et al.*, 2014).

## 2.4 Feature counting

After the reads are aligned to the genome, the next step is to count how many reads have mapped to each gene. Counting techniques are categorized into Count-based quantifiers and Poisson model-based quantifiers.

### 2.4.1 Count-based quantifiers

The count-based models assume that all the reads map uniquely to the genome and therefore  $T$  is the set of reads that has a length  $l_t, t \in \rho = \{\rho_t\} t \in T$  is defined to be the relative abundance of reads such that  $\sum_{t \in T} \rho_t = 1$ .  $F$  denotes the set of single end reads and  $F_t \subseteq F$  the set of reads mapping to the genome  $t$ . The assumption here is that all the reads in  $F$  have got the same length  $m$ . Note that in genome  $t$ , the number of positions in which a read can start is  $\tilde{l}_t = l_t - m + 1$ . The adjusted length  $\tilde{l}_t$  is called the effective length of  $t$ . In the generative model, first a transcript is chosen from which to select a read  $f$  by

$$\mathbb{P}(f \in t) = \frac{\rho_t \tilde{l}_t}{\sum_{r \in T} \rho_r \tilde{l}_r} \quad (2.2)$$

Next, a position in that transcript is selected uniformly at random from among the  $l_t - m + 1$  positions. Thus, the likelihood of observing the reads  $F$  as a function of the parameters  $\rho$  is  $\mathcal{L}(\rho) \prod_{t \in T} \prod_{f \in F_t} \left( \frac{\rho_t \tilde{l}_t}{\sum_{r \in T} \rho_r \tilde{l}_r} \cdot \frac{1}{\tilde{l}_t} \right)$  (2.3)

The expression profile is the accumulated read count on each targeted gene, and each count-based quantifier uses a proprietary filtering criterion (Kanitz *et al.*, 2015).

### 2.4.2 Poisson model-based quantifiers

This model assumes that for a set of aligned fragments  $F$ , for every  $s \in U$  the number of reads starting at  $s$  is Poisson distributed with rate parameter:

$$\lambda_s = \sum_{k=1}^k c_{s,k} \frac{\kappa_k}{L_k}, \quad (2.4)$$

Where  $\kappa_k$  is a rate parameter for transcript  $k$ . Here  $C = \{c_{s,k}\}_{k=1, \in U}^k$  is a site-transcript compatibility matrix with  $c_{s,k} = 1$  if transcript  $k$  appears in some element of  $s$ , and 0 otherwise.

## 2.5 Feature Selection Methods

Reduction in the initial features to a smaller subset that has got enough information to represent the entire dataset and provide better results of the machine learning models have been regarded to as feature selection (Mehmood *et al.*, 2019).

### 2.5.1 Filter-based Methods

A filter-based feature selection methods work by selecting the most important features from the original list by putting into consideration statistical characteristics of the features that have been provided. After the selection only the significant features are provided as input for the learning model that produces the output. This process improves the process of prediction as well as classification accuracy as well as reduced computation time and the problem of overfitting. Since this feature selection technique considers statistical relationship only, it is faster than wrapper methods and very suitable in dimensionality reduction. Filter techniques like information gain, (IG), Gain Ratio (GR), Chi-squared (CS), Relief-F (RF) and Minimum Redundancy Maximum Relevance, (mRMR) have been ranked as best alternatives in handling high dimensional data because of the simple ranking strategies that are applied by these algorithms (Bommert *et al.*, 2020). IG, GR, CS, have also been reported to provide improved classification accuracy as long as the most irrelevant features are removed from the entire dataset by use of a statistical ranking scores and a set of threshold values that have been defined by the user. Significant features are those that have the highest-ranking scores above the given threshold values, whilst features with lower ranking scores are deleted and not included during the classification

phase (Alirezanejad *et al.*, 2020). Ali *et al.*, (2019) indicated in their study that the accuracy of classifiers like the widely used support vector machines are affected in terms of performance anytime the number of features that are selected from each filter-based algorithm are either excessively large or extremely small. This imbalance problem is usually caused by the fact that each independent filter-based algorithm focuses on evaluation of each feature separately instead of putting into considering other factors such as interactions or dependencies between or among features. This working principle makes them fail to produce optimal number of features that are appropriate for classification task which makes the classifiers to perform relatively poor (Ali *et al.*, 2019).

Thakkar & Lohiya, (2021) did a study for analyzing the effect associated with feature selection techniques based on detection rate as well as accuracy of the system. They measured accuracy, precision, recall, and f-score first using all features of the dataset and another experiment using only features selected by three filter-based feature selection algorithms which are Chi-Square, IG, and Recursive Feature Elimination (RFE). They later did an evaluation of every class individually since they exhibited dissimilar characteristics. The comparative analysis of various classifiers showed improvement in model performance when filter-based feature selection methods were incorporated in the model (Thakkar & Lohiya, 2021).

### 2.5.1.1 Chi-Squared statistics (X2)

This is a univariate filter method that is built on the  $\chi^2$  statistic that does the evaluation of every feature individually with respect to the classes they belong to. Relevance of the features with respect to their class is based on the highest Chi-square value. With a number of intervals ( $V$ ), classes number ( $B$ ), and instances total number being ( $N$ ), then the Chi-squared value for every feature in a class is calculated using equation 2.5 below:

$$x^2 = \sum_{i=1}^V \sum_{j=1}^B \frac{\left( A_{ij} - \frac{R_i * B_j}{N} \right)^2}{\frac{R_i * B_j}{N}} \quad (2.5)$$

Where  $R_i$  represents the instances number within the range  $i$ th ,and  $B_j$  is the number of instances in the class  $j$ th and  $A_{ij}$  the number of instances in the range  $i$ th and class  $j$ th .

Chi-square has been used in several studies on different types of datasets. Şahin *et al.*, (2021) did a classification of microstructure images dataset that used an improved wrapper-filter based feature selection technique by use of texture-based feature descriptor. A feature descriptor known as rotational local tetra pattern (RLTrP) was used in extraction of relevant features from the input images before feature selection. This was then followed by an ensemble of three filter-based methods which was generated by the top-n features that were selected by Chi-square, Fisher score, and Gini impurity-based filter methods. The study's major goal was to combine various filter-based methods to extract features that would be used to populate a wrapper-based meta-heuristic feature selection algorithm known as harmony search (HS). When determining the fitness value, the author's defined HS using the objective function of Pearson correlation coefficient and mutual information. The authors reported optimized features with less dimension and improvement in classification accuracy of the seven-class microstructural images (Şahin *et al.*, 2021).

Sridhar & Sanagavarapu (2021) accessed information from DarkNet, a database of encrypted information from websites that host illicit activity and secret services. The internet activity on these privatized networks is anonymous and nearly untraceable. DarkNet which was a traffic classification proposed in this study helped in improving network security by detecting dangers or risks to any systems of network. CIC-Darknet2020 dataset was employed in their research and to aid in feature selection, a feature importance analysis was done on the dataset using a Chi-Square statistical score. Random Forest classifier has been used to produce a multi-class classification in traffic encryption categorization, with an F1-Score of 97.87 (Sridhar & Sanagavarapu, 2021).

In determining features, Rahman & Mahmood (2022) integrated the Chi Square Test and Pearson's Correlation Heatmap. After that, the most relevant output from the classification algorithms, which included KNN, SVM, and Decision Tree algorithms, was calculated using the stacking ensemble technique. They also used three boosting strategies to obtain the output, as well as voting ensemble methods. After preprocessing the data, the Cleveland datasets were imported into the constructed model, and the two feature selection methods, Chi-square test and Pearson's Correlation, were applied. The Chi-square test produced six essential features from the dataset, according to the researchers, which were based on the top rank (Rahman & Mahmood, 2022)

Mehmood *et al.*, (2021) conducted a study on the efficient and smart use of electrical energy in residential and commercial buildings, which necessitated a complete examination of energy usage across all equipment. They based their research on IEEE standard 1459, which uses voltage and current signals from distinct appliance events that are on or off to determine different power quantities. The most relevant collection of features was chosen using three feature selection algorithms which were neighborhood component analysis (NCA) MRMR and Chi-square tests. The selected features were thereafter used as input for categorizing appliances into pre-defined categories. The authors claimed that their method reduced processing time and improved classification accuracy (Mehmood *et al.*, 2021).

### **2.5.1.2 Information Gain**

Information gain (IG) is a feature selection method that determines the relevance and calculates the information gain ratio between features and their class labels to determine the dependency degree of features (Fahrudin *et al.*, 2016). In order to obtain the ranking score, IG calculates the entropy value for each attribute as well as its relevance score.

Information gain which is the highest corresponds gaining a substantial information gain, the lowest value of entropy is regarded relevant. This implies that if there is a decrease of entropy value, then its and indication that the information was obtained based on new information that has been added. IG, on the other hand, favors features having a large number of different values (Ab Hamid *et al.*, 2021). Therefore, IG can lead to overfitting problem since it cannot handle features that are redundant. IG filter has been regarded as one of the most commonly used univariate methods of evaluation. This filter considers only one feature at a time, filtering a feature based on its information gain. This method uses entropy as a criterion for ranking variables. The entropy of a class feature  $Y$  is defined as:

$$H(Y) = -\sum p(y) \log_2(p(y)), \quad (2.6)$$

where  $p(y)$  is defined as the marginal probability of the density function variable  $Y$  which is random. When there is partitioning of the values of  $Y$  that have been detected, then  $S$  which is the dataset for training is divided based on the second feature  $x$ . Therefore, the entropy of  $y$  in relation to the partitions that have been induced by  $x$  becomes less than the entropy of  $Y$  before partitioning. Therefore, there exist a relationship between features of  $y$  and those of  $x$ . So, the entropy of  $y$  after observation of  $x$  is:

$$H(Y|X) = \sum p(x) \sum p(y|x) \log_2(p(y|x)), \quad (2.7)$$

with  $p(y|x)$  being the conditional probability of  $y$  when given  $x$ . If the entropy is provided as a criterion of “impurity” in a set of training data  $S$ , the measure to reflect extra information about  $Y$  when  $X$  is provided can be to represent the amount by which the entropy of  $Y$  decreases. This measure is an indication of the dependency that exist between  $X$  and  $Y$ , which is called IG whereby:

$$IG = H(Y) - H(Y|X) = H(X) - H(X|Y). \quad (2.8)$$

IG is regarded as a symmetrical measure. This technique works by providing an ordered features for classification with a threshold for selection of the required number based on

the order obtained. A main drawback associated with IG criterion is biasness and favoring of features that have more values even when they are not informative.

Several studies in data mining have used Information gain as a feature selection technique. One of the earliest studies on sentiment analysis was done by Mukras *et al.*, (2007). In this study, they investigated information gain accuracy as feature selection method. They reported that identification of discriminatory features was not possible with information gain. To overcome this problem, they proposed another approach called a probability redistribution procedure (PRP). There was an improved classification accuracy that was reported using PRP approach (Wu & Xu, 2015).

Nimbalkar & Kshirsagar, (2021) proposed an approach of selecting features for intrusion detection systems (IDSs) by use of Information Gain (IG) and Gain Ratio (GR) with the highly ranked top 50% features for the detecting both denial of service attacks as well as distributed denial of service attacks. Their approach obtained subset of features by use of insertion and union operations on the subsets that was obtained to 50% features that were ranked by both feature selection techniques (Nimbalkar & Kshirsagar, 2021). Saheed & Hambali, (2021) did a study on customer churn which is an important issue and worries large businesses operating online shops. Customer churn prediction for variables identification that are potential contributors to customer turnover is an important step in reducing churn. Therefore, in their study, they created a churn prediction model using machine learning approaches like the Support Vector Machine (SVM), the Multi-Layer Perceptron (MLP), the Random Forest (RF), and Naïve Bayes (NB). The feature selection that was used in this study before creating the prediction model was a combination of IG combined with Ranker based methods. The performance of the model for this study was evaluated using measures such as the accuracy measure, precision measure and F-measure combined with 10-fold cross-validation. Authors reported an



accuracy of 95.02% after feature selection was done and an accuracy of 92.92% before performing feature selection (Saheed & Hambali, 2021).

Fahrudin *et al.*, (2016) used entropy formula in selecting the best attributes from the original data. Clustering algorithm was later used in getting the number of attributes that could be filtered by information gain from the ranking attributes. This clustering algorithm that was used in their study hierarchical K-means (K-means optimization) that categorized cancer patients as either normal or cancerous. Their experiments showed that the information gain technique had selected 12 of 18 attributes with the highest contributing factor to the breast cancer patient's treatment that was based on the last condition. There was a slight decrease in the clustering algorithm error ratio that was reported by authors ranging from 44.48% (using 18 initial attributes) to 21.42% (Fahrudin *et al.*, (2016). Another study by Pati, (2018) used IG as feature selection approach in cancer prediction followed by an advanced machine learning technique which was used to find maximum probability of cancer-causing genes. Salem *et al.*, (2017) proposed an ensemble method that combined IG and SGA algorithms in classification of human cancer diseases. This method used IG for selecting the features followed by feature reduction by GA and finally Genetic Programming (GP) was used to classify types of cancer. This approach however had a limitation associated with time complexity.

### **2.5.1.3 Relief-F**

ReliefF is a filter-based algorithm which handles multiclass data challenges, and it is regarded as more robust and has capability to deal with incomplete and noisy high dimensional data. ReliefF works by making a random selection on an instance of  $R_i$  from the provided data that must be analyzed and all the available k-nearest neighbors who are from the same nearest hits, ( $H_j$ ) is placed based on the class and its nearest neighbors from every other different class considering the nearest misses  $m_j(C)$ . The estimation quality  $W(A)$  for all the attributes  $A$  depends on their  $R_i$ , hits  $H_j$  and  $m_j(C)$  misses. In case the

instances of  $R_i$ , hits  $H_j$  have got different values of the attribute  $A$ , then this attribute makes the separation of the same class, which is not appropriate leading to decrease of quality estimation  $W(A)$ . On the contrary, when instances of  $R_i$  and  $m_j$  have got different values of the attribute  $A$  for a certain class, then the attribute  $A$  makes a separation of the two instances with different class values desirable making  $W(A)$  which is the quality estimation increase. Since Relief-F takes into consideration the problem of multiclass, then the average of all hits and misses are done. Also, the contribution for every class of the misses is usually weighted using estimates from the prior probability of that given class  $P(C)$ . The entire process is repeated  $t$  times, with  $t$  being a parameter that is defined by the user (Bolón-Canedo, 2014).

Relief-F is regarded as a nearest-neighbors feature selection approach because of its ability in identification of statistical interactions among features in high dimensional data. This algorithm has been widely used to identify effects of gene-gene interaction in both simulated and real genome-wide association studies (Urbanowicz *et al.*, 2018). Relief-F uses a function called a “*diff*” which determines the available nearest neighbors within the space of single nucleotide polymorphisms (SNPs) and then to compute the importance of every SNP based on its ability in separating treatments and controls within SNP space (Arabnejad *et al.*, 2018). ReliefF has been widely applied in feature selection across all domains. Sadiq *et al.*, (2021) did a study to reveal brain connectivity patterns and applied feature selection algorithm ReliefF and Pearson's correlation connectivity (PCC) to distinguish diseased samples of patients with Alzheimer's disease from samples of normal controls. PCC measures the correlation between specific regions whereas ReliefF is well known technique in handling high dimensional data feature vectors and they combined both techniques. the authors reported a classification accuracy of 93.5% using a k-nearest neighbor (KNN) classifier (Sadiq *et al.*, 2021). Angadi & Reddy, (2021) used ReliefF feature selection approach after feature extraction in sentiment analysis study to pick optimal features. This was followed by classification using random forest classifier to categorize sentiments of speakers as either neutral, positive or negative class. The authors reported that the quantitative analysis of the proposed approach enhanced the

classification accuracy up to 5.41% as compared to the existing systems (Angadi & Reddy, 2021).

Ali & Baiee, (2021), did a study in identifying a subset of attributes from the Queensland roads dataset using multiple feature selection methods which were IG, GR, Chi-Squared and also Relief-F). A comparison on the evaluation results among these feature selection methods showed that features from applying Relief-F resulted into a highest classification accuracy with 80.39% over other feature selection methods classified using artificial neural networks (Ali & Baiee, 2021). Chen *et al.*, (2022) used relief-F algorithm in reduction of the dimensions of feature vector feature selection and optimization for reducing the dimensionality with an aim of reducing system calculation complexity that ensured accuracy. Finally, using the ranking feature vectors as input, a classifier based on support vector machines (SVM) was created, and the authors reported excellent classification accuracy (Chen *et al.*, 2022).

ReliefF has been associated with good performance particularly on the microarray data sets by providing highest test set accuracy on data sets (Alhenawi *et al.*, 2022). Other than dermatology data set, all other datasets showed significant improvement in accuracy for the feature selection techniques as compared to test accuracy without feature selection done on data. For the KNN classifier, the similarity classifier included all features in the classification model, which made it susceptible to irrelevant as well as noisy features. Therefore, by the fact that feature selection in most cases has been associated with improved accuracy, this classifier is intuitive (Alhenawi *et al.*, 2022). Alsahaf *et al.*, (2022) proposed a feature selection-based technique called FeatBoost and in most datasets used in their study, the authors reported superiority when compared with Boruta and Relief-F but when features are fewer indicating that smaller subsets of data tend to have relevant. For the computation time, XGBoost was reported to be the most effective

technique across all datasets used in their study seconded by ReliefF (Alhenawi *et al.*, 2022).

#### 2.5.1.4 Minimum Redundancy Maximum Relevance (mRMR)

mRMR is a method for feature selection which eliminates redundant and irrelevant features automatically in a high-dimensional feature space and selects only informative features based on the criteria of maximum correlation and minimum redundancy. mRMR approach is regarded as multivariate filter method which selects only features of the highest relevance and minimally redundant to the target class meaning that features that are similar to each other are selected (Hu *et al.*, 2020). Being a multivariate filter method, feature dependencies are modeled, and redundant features are detected. mRMR approach puts into consideration features relevance and the redundancy of the feature in respect to target class. Features are considered as relevant if there is best trade-off between maximum relevance to the target as well as minimum redundancy. The working principle of this approach makes it less scalable and slower when compared with univariate techniques (Bolón-Canedo, *et al.*, 2016). Mutual information finds a set of feature  $S$  which has  $m$  features  $\{x_i\}$ , and both features have got highest scores based on the target class  $Y$ . This is known as maximum dependency, described as:

$$\max D(S, Y), D = I(\{x_i, i = 1, \dots, m\}; Y). \quad (2.9)$$

whenever  $m$  is equal to 1, solution becomes the features that maximizes  $I(x_j; Y) (1 \leq j \leq M)$ . When  $m > 1$ , this is followed by an incremental search scheme which adds every feature at a time when given the set with  $m - 1$  features,  $S_{m-1}$ , then  $m$ th feature is determined such that the feature that makes the largest contribution to the increase of  $I(S; Y)$ , that takes form:

$$I(S_m; Y) = \iint p(s_{m:Y}) \log \frac{p(s_{m:Y})}{p(s_m)p(Y)} dS_M dy \quad (2.10)$$

Despite the theoretical significance of maximal dependency, getting an accurate estimate of density for multivariate data is typically difficult  $p(x_1, \dots, x_m)$  and  $p(x_1, \dots, x_m; Y)$ , due to two high-dimensional space difficulty which are insufficiency of total number of samples and computation of high-dimensional covariance matrix inverse which a challenge. Another drawback associated with maximum dependency is the low computational speed. These challenges are prominent in both continuous and discrete variables (Bolón-Canedo, *et al.*, 2016). For example, if every feature in  $N$  samples have three categorical states, then  $K$  features can have a maximum of  $\min(3^k; N)$  joint states. A quick increase in joint states as compared to the samples number  $N$  then a problem of correct estimation of joint probability and mutual information becomes inherent. Therefore, even though *maximum dependency* selects a relatively small number of features, with large  $N$ , this approach is not the best when the aim of experiment is to get high classification accuracy since *maximum dependency* approach is difficult to implement, the alternative is selecting features based on *Maximum relevance* criterion. Maximum relevance searches for features that satisfies the equation below which does the approximation of the *max-dependency* using mean values of all mutual information of the values between each individual feature  $x_i$  (of set  $S$ ) and class  $Y$

$$\max D(S, Y), D = \frac{1}{|S|} \sum_{x_i \in S} I(x_i; Y) \quad (2.11)$$

A major challenge associated with maximum relevance is that the selected features can be highly redundant with dependency among the features being large. Whenever two features depend highly on one another, the discriminative power of the class does not change much even when one of them is deleted. Min-Redundancy condition is therefore added to select features that are mutually exclusive as shown in the equation below.

$$\min R(S), R = \frac{1}{|S|} \sum_{x_{i,x_j} \in S^1} (x_{i,x_j}) \quad (2.12)$$

Minimal-redundancy maximal- relevance is the name given when the above two constraints are combined (Toğaçar *et al.*, 2020).

Li *et al.*, (2018) used mRMR approach in eliminating redundant features and selected the first vital features that were used for constructing new fault feature vectors for representing characteristics of faults. Validation of mRMR method, which offers faster calculation and robustness, was done by Yan, *et al.*, (2019). mRMR feature selection and grid search support vector machine was used in mechanical components fault identification (Yan *et al.*, 2019).

Chen *et al.*, (2022) proposed a model for diagnostic that was based on convolutional neural network (CNN, local interpretable model agnostic annotations (LIME) as well as mRMR methods. Their model was meant to detect four different types of white blood cells. To determine vital regions of the images to be used for classification SqueezeNet and mRMR feature selection algorithm were used to extract features from the images. The obtained feature sets were combined with LIME algorithm and classified using support vector machines. The authors reported an accuracy of 95.88% for the proposed model therefore selecting features with SqueezeNet and mRMR technique coupled with support of LIME affected the model performance positively (Chen *et al.*, 2022). Jo *et al.*, (2019) Applied Pearson's correlation coefficient as a measure of redundancy and R-value as a measure of relevance. Then later compared the original mRMR and their proposed method by selecting features using both methods on various datasets followed by the classification

test. The authors reported higher accuracy in their proposed approach as compared to original mRMR (Jo *et al.*, 2019).

### **2.5.2 Wrapper-based Methods**

When picking a subset of features, wrapper techniques use a learning algorithm as part of the evaluation function. To guide the search, this approach uses a black box unlike other approaches that use entropy or sufficiency of the subset. An evaluation function for every candidate feature subset gives an estimation of the model quality based on the learning algorithm induction. This leads to increase in computational time because every candidate feature subset must be evaluated when doing the search, and the target learning algorithm must be done many times like ten-fold cross-validation which is used in model quality evaluation (Bolón-Canedo, 2014). Hameed *et al.*, (2018) in their study did an evaluation of wrapper-based methods in which unlike filter methods, there was possible communications among the variables. The authors reported that the best subset that had the highest accuracy to model was achieved by wrappers. Wrapper-based approaches result into few numbers of features which have a robust discriminative power. In another study a hybrid in form of filter and wrapper, that consisted of information gain as well as a standard genetic algorithm was used in feature selection and classification (Zhao *et al.*, 2019). Additionally, wrappers are classifier dependent meaning that similar results are not certain when a different classifier is applied. Therefore, it is recommended that when wrapper method is used, then different classifiers should be applied for the purpose of feature selection (Hameed *et al.*, 2018).

### **2.5.3 Embedded methods**

Embedded method is a feature selection approach with working principle just wrapper approaches, since they also depend on a given learning algorithm. However embedded methods are less computationally intensive as compared with wrapper when interacting with the classifiers. Hence, the embedded methods combine the filters efficiency with the accuracy of wrappers. Their implementation is based on the feature selection that are built-

in which is performed by the feature reduction. Two major examples of the embedded systems are LASSO and RIDGE regression feature selection algorithms (Hameed *et al.*, 2018). Other studies that have used wrapper methods are for example a study by Salekin & Stankovic, (2016) introduced a machine learning based wrapper method to identify a set of twelve attributes. Ranking of attributes was done based on a prognostic potential in detecting (CKD) followed by attributes reduction to ten using LASSO regularization method. Some authors have reported improvement in accuracy of 0.993 together with root mean square error of 0.1084. Guo *et al.*, (2019) proposed an embedded feature selection algorithm called ensemble embedded feature selection (EEFS). This approach was capable of more effectively and efficiently addressing a multi-label bioinformatics data learning challenge. The authors reported that this proposed algorithm could reduce data errors that are accumulated through application of an ensemble method. Their experimental results were obtained from five multi-label bioinformatics datasets (Guo *et al.*, 2019).

Feature selection is an important phase during variable prediction in an industrial production according to (Li *et al.*, 2021). In their research, they proposed an embedded feature selection method based on vector machine relevance and marginal approximation of the likelihood function. Hierarchical prior distributions were established, and the joint posterior distribution over the model weights and kernel parameters was calculated using a Gibbs sampling method combined with a Laplace approximation. As a result, feature selection was done by looking at the posterior of the kernel parameters. Two industrial data sets were used to test the performance of their suggested technique. The authors reported improved prediction accuracy of their model. a series of benchmark datasets and two practical industrial datasets are employed (Li *et al.*, 2021). According to Deng *et al.*, (2019) Filter model has been regarded as the most efficient approach whose investigation has been extensive in text categorization. However, the use of wrapper and embedded methods is limited in text categorization because of computation cost when working with a text document that contains a lot of features. This challenge has been addressed by use



of hybrid techniques where filter methods are used in eliminating redundant and irrelevant features and the selected feature in turn is fed to a wrapper method for further processing (Deng *et al.*, 2019).

## **2.6 Feature Extraction methods**

Feature extraction is one of the dimensionality reductions approaches that handles the problem of obtaining the most informative features of a given problem to improve data storage or processing efficiency. The two stages of feature extraction are feature creation and feature selection. To transform "raw" data into a set of useful or meaningful features. It can be thought of as a preprocessing transformation that includes standardization, normalization, discretization, signal augmentation, and local feature extraction, among other things. Principal Component Analysis is the most widely used feature extraction approach (Liu & Motoda, 2007).

### **2.6.1 Principal Component Analysis (PCA)**

PCA is a dimension reduction approach that works by identifying significant data from large data sets. The goal of PCA is to reduce the original dataset to a smaller feature set with fewer dimensions. PCA relies heavily on determining the number of major components. The principal components that best represent the data should be the  $p$  number of principal components chosen from all principal components. Some of the criteria used to identify the appropriate number of principal components include the broken-stick model, cross-validation, Velicier's partial correlation process, Kaiser's criterion, Barlett's test for equality of eigen-values, Cattell's screen-test, and cumulative percentage of variance. (Shah & Patel, 2016). The primary goal of PCA is to find an appropriate linear combination of the data matrix  $X$ . The Jordan decomposition of the covariance matrix of  $X$  is used to achieve this goal of the covariance matrix  $\Sigma$  of  $X$  (also the correlation matrix  $S$  of  $X$ ). The random vector contained in the data matrix  $X$  is represented as  $X_{i \times p} = (x_1, x_2, \dots, x_p)$

with mean  $\mu_i \times p$  and covariance matrix  $\Sigma$ . A transformation as shown in the equation below is the key component of the PCA:

$$x_{i^*p \rightarrow y_{i^*p}} = (x - \mu)_{i^*p} \Gamma_{pxp} \quad (2.13)$$

Where  $\Gamma$  is attained using Jordan decomposition of  $\Sigma$ ,  $\Gamma^T \Sigma \Gamma D = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_p)$  with  $\lambda_i$ 's being the decomposition eigen values. Each element of  $y_i \times p$  is elements combination of a linear representation of  $x_i \times p$ . Furthermore, each element of  $y$  is independent of the others. As a result, we get  $p$  main components which are the  $p$  eigenvalues of the Jordan decomposition  $\Sigma$ . In general, the initial principal components are used for further analysis (Dua & Chowriappa, 2012).

### **Assumptions of PCA.**

A few conditions must be met to get relevant results from PCA. Because the typical PCA explores the covariance/correlation patterns, which makes sense only for the selected variables, the input data must first be continuous variables of real value, evaluated on an interval scale or ratio (Todorov *et al.*, 2018). For discrete variables which are measured on an interval scale like integers or categorical variables, correspondence analysis, or non-metric multidimensional scaling are all viable methods. Second, the linearity of the link between each pair of variables is required by the covariance/correlation measures. When nonlinear relationships are discovered, data transformation techniques such as logarithmic transformation should be examined. Outlier detection is required prior to analysis as atypical values can mislead the results by affecting the amount of the covariance / correlation (Tabachnick *et al.*, 2018).

PCA has been used to reduce dimensionality in a variety of fields of research. Mallick *et al.*, (2021) proposed a new paradigm for microarray data classification. Ant Colony Optimization (ACO) is used to modify the parameters of an ANN. For dimensionality reduction, PCA was performed, and the reduced dataset was optimized by Ant Colony Optimization (ACO) in the first phase followed by training with Functional Link Artificial

Neural Network in the second step (FLANN) (Mallick *et al.*, 2021). A comparison of RF, RF-PCA, as well as a multi-class tumor classification approach using RF classifier, was presented by Saraswathi & Gupta (2019). According to the experimental results, the random selection of RF-PCA provided higher accuracy than other techniques. In addition, Jamal *et al.*, (2018) looked at how feature extraction can reduce the number of features that were required to classify breast cancer using original white blood cells data set. Dimensionality reduction utilizing the K-means cluster was virtually as excellent as PCA, according to the metric assessment (Saraswathi & Gupta, 2019). Salo *et al.* (2019) developed a new hybrid strategy that combines Information Gain (IG) and PCA to eliminate extraneous features while retaining the best attribute subset. The proposed approach's resilience showed good results in both the NSL-KDD and Kyoto 2006+ datasets. To extract useful discriminative properties, Bossaghzadeh, (2020) did their study on Hoda dataset using a fine-tuned deep Neural network. The attributes were used to classify the data using a linear SVM. In the second experiment, they used PCA to decrease the extracted feature measurements to enhance accuracy and reduce processing burden, and then submitted the data to a Support Vector Machine (Bossaghzadeh, 2020).

Kim *et al.*, (2018) merged several omics datasets and offered two forms of PCA meta-analysis frameworks, namely, Meta PCA. Three meta-analysis transcriptional investigations in the yeast cycle, prostate cancer, rat metabolism, and a pan-cancer methylation research from The Cancer Genome Atlas were created using simulators (TCGA) were used in this study. As a result, the proposed structure's detailed visualization, resilience, and discovery were improved. Yang *et al.*, (2018) used a principal component analysis network (PCANet) to extract features from a noisy EigenECG Network (ECG) signal and linear SVM was used to improve the speed of classification. They developed five types of imbalanced starting and noise-free conditions for testing the effectiveness of their approach. Using waters of China's Lake Nancy Basin data set, Xu *et al.*, (2021) Principal Component Analysis was used in a Fuzzy Comprehensive Evaluation (FCE-PCA). By generating organic functions through these

semi-sinusoidal distribution systems, measuring weight using several additional standard methods, solving self-equation using Jacobi, and removing the main components based on inherent values, the percentage of the contribution accumulated, and packing, the efficiency of extraction of main contaminants was improved. Raunak, (2017) disclosed a new strategy for constructing lower-dimensional word embedding that effectively combines the reduction of PCA-based dimensionality with a previously described post-processing algorithm. Empirical studies on 12 typical word similarity benchmarks demonstrated that their technique decreases the embedding's dimensionality by half, resulting in comparable or (more often) superior efficiency than the higher-dimensional embedding (Raunak, 2017).

Kaya *et al.*, (2017) investigated the efficacy of PCA clustering based on brain tumor images. The PCA method was first used to analyze MRI images of various sizes in this model, followed by clustering using K-means and FCM. A greater performance rate is achieved by combining PCA and K-means. To categorize intrusion detection datasets, Bhattacharya *et al.* (2020) introduced a PCA-Firefly approach. To perform data transformation one-hot encoding technique was used, and the dimensionality reduction was done using the PCA-Firefly technique. The XGBoost classifier is then used to classify the dimensionally reduced dataset. The superiority of this proposed model was established by experimental results (Bhattacharya *et al.*, 2020).

Gadekallu *et al.* (2020) introduced a PCA-Firefly based Deep Learning model that was used for early diabetic retinopathy detection. A PCA-Firefly algorithm could select the best features, then Deep Neural Networks was used to classify the diabetes retinopathy dataset. Authors reported improved classification results as compared to other machine learning algorithms (Gadekallu *et al.*, 2020). In study by Fujisawa *et al.*, (2021) RNA expression profiles of 16 COVID-19 patients and 18 healthy control subjects were evaluated using unsupervised feature extraction based on principal component analysis. (PCAUF). From 60,683 potential probes, 123 genes were identified as crucial for COVID-19 development, including immune-related genes. The authors did a patient/non-patient categorization based on the identified genes to ensure that the genes chosen by the

proposed model were effective for the diagnosis of COVID-19 patients. Classification models such as logistic regression (LR), support vector machine (SVM) and random forest were used after feature selection (Fujisawa *et al.*, 2021).

Since the biological signatures of two of the most common subtypes, lung adenocarcinoma (LUAD) and lung squamous cell carcinoma (LUSC), are different, Chen & Dhahbi (2021) took advantage of the fact that they are generally treated identically and lumped together as non-small cell lung cancer (NSCLC). The biomarkers LUAD and LUSC are uncommon, and their biological mechanisms are still unknown. To detect physiologically significant signals, many studies have attempted to improve existing machine learning algorithms or develop unique biomarker discovery methods. However, for cancer classification, biomarker discovery, or gene expression analysis, few studies have employed overlapping machine learning or feature selection methods. Their study recommended use of overlapping classical feature selection or feature reduction strategies. Genes that had been chosen using the overlapping technique were then verified using random forest. The overlapping method's classification statistics were compared to those of classic feature selection methods. AUC and ROC analyses were used to validate the biomarkers in an external dataset (Chen & Dhahbi, 2021).

Roy & Taguchi (2021) applied unsupervised feature extraction (FE) based on PCA, tensor decomposition (TD), and kernel tensor decomposition (KTD) to a hypoxic data set. Authors discovered that unsupervised FE based on PCA, TD, and KTD could successfully identify a small number of genes linked to changed gene expression and m6A profiles, as well as the enrichment of hypoxia-related biological words, with enhanced statistical significance (Roy & Taguchi, 2021). Bai & Hira (2021) proposed a method for categorizing cancer data as follows: a) Using various feature selection techniques, such as principal component analysis (PCA), chi-square, genetic algorithm (GA), and F-score,

were extracted information from a larger dataset; b) using majority voting ensemble SVM, authors classified extracted information into normal and malignant classes samples. Different SVM kernels, such as linear, polynomial, radial basis function (RBF), and sigmoid, were used in SVM ensemble-based technique. A majority voting strategy was used to integrate the estimated results of specific kernels. Using ensemble SVM classification, the algorithm's performance is validated on six benchmark cancer datasets: colon, ovarian, leukemia, breast, lung, and prostate (Bai & Hira, 2021).

Among other drawbacks, the PCA result is frequently uninterpretable on its own and this calls for the combination of the various algorithms since different approaches choose features based on different criteria. Since each approach has its own set of strengths and weaknesses, focusing on the overlapping features will maximize the strengths and reduce the flaws of each method, lowering the number of false positives and increasing the reliability of the results (Dhahbi, 2021).

### **2.6.2 Partial Least-Squares-Based Dimension Reduction (PLS)**

PCA uses an unsupervised approach to determine the linear connection between variables. However, it is occasionally desirable to determine the degree of dependence between variables while also considering the goal variable. One such dimensionality reduction technique is partial least squares (PLS), which was first developed as a matrix decomposition technique before being used as a multivariate regression tool. PLS, on the other hand, has lately been discovered to be a successful dimension reduction approach. PLS is founded on the idea that seen data is generated by a system or process that is driven by a small number of latent (non-observable or measured) attributes (Khare *et al.*, 2019). As a result, the goal of PLS is to find uncorrelated linear transformations (latent components) of the original predictor characteristics that have a high covariance with the response features (Khare *et al.*, 2019).

Vanitha *et al.*, (2015) developed a classifier model utilizing PLS and the Ridge Penalize Logistic Regression regularizing method to minimize the microarray data dimension. RPLS was the name of the model developed, and it was able to deliver a low level of classification error. However, the problem with the statistical approach is that it has a rigid classification system, making it difficult to categorize a sample if the gene expression of the sample differs somewhat from the gene characteristic that has been determined. Another recent study by Chen *et al.*, (2021) analyzed differences between lung cancer and glioma and proposed a method that was based on patient serum Raman spectra in combination with deep learning. According to their findings, the effect of PLS feature selection on classification was better than experimental results, the classification effect of PLS after dimension reduction is substantially better than that of PCA, most probably because PCA retains substantive number of components that had noise. PLS can reduce meaningless noise and make the model contain the fewest variables possible to produce relatively optimal lowdimensional data. However, the major drawback of this approach is that it can only fit linear classification problems and is prone to the over-fitting problem (Wang *et al.*, 2020).

### 2.6.3 Factor Analysis (FA)

Factor analysis (FA) is a linear method just like PCA. FA hypothesizes that the measured variables are influenced by a set of unknown and frequently unquantifiable common causes. FA is motivated by the need to discover hidden relationships, and it can thus be used to reduce the dimension of datasets utilizing the factor model. According to the  $k$ -factor model, a  $p$ -dimensional random vector  $1xp$  with covariance matrix  $\Sigma$  satisfies the  $k$ -factor model if

$$x = \Lambda f + u \tag{2.14}$$

where  $Ap \times k$  is a matrix of constants,  $fk \times 1$  represents random common factors, while  $up \times 1$  represents a specific factor. Furthermore, the factors are all uncorrelated in the k-factor model, and the common factors are normalized (Ghojogh *et al.*, 2021).

#### 2.6.4 Linear Discriminant Analysis (LDA).

LDA is a discriminant approach that tries to model differences between samples assigned to different groups. The method's goal is to optimize the between-group variance to within-group variance ratio. When this ratio reaches its greatest value, the samples within each group have the smallest possible scatter and the groups are the most distanced from one another. For a two-class discriminant issue, once the LDA assumption of equal group covariance's is met, the expression is maximized.

$$S = \frac{p^C b p^T}{p^C w p^T} \quad (2.15)$$

where  $C_b$  and  $C_w$  are the between- and within-group covariance matrices, and  $p$  is the multivariate data space direction that best separates the two groups of samples. It is vital to note that  $p$  is the eigenvector derived from the PCA decomposition of matrix  $C_w - 1C_b$  at this point. A multi-class problem can be generalized from the two-class discriminant problem. Because LDA is based on traditional estimators of location and covariance, it is sensitive to outlying samples, with an increase in the number of erroneously allocated samples lowering LDA performance. Using robust estimators of data location and covariance instead of their classic counterparts is a reasonably simple way to address the LDA's lack of robustness (Sarraf & Pattnaik, 2020).

Reddy *et al.*, (2020) did a study using Diabetic Retinopathy (DR) and Intrusion Detection System (IDS) datasets to examine the performance of PCA and LDA. PCA reduced the features to 26 from 36 dependent attributes while keeping 95% of the dataset, while Linear LDA reduced the features to 1. Experiments show that when the dimensionality of the



datasets is high, ML techniques using PCA yield better results. When datasets have a low dimensionality, it has been observed that ML methods without dimensionality reduction produce better results. It is evident from the preceding discussion that the goal of LDA is not to reduce dimensionality (Reddy *et al.*, 2020). Findings of the study done by Hasan & Abdulazeez, (2021) discovered that classifiers with PCA perform better than those with Linear Discriminant Analysis (Hasan & Abdulazeez, 2021).

## 2.7 Graph Definition and origin of Graph Theory

A graph is a collection of points with lines that connect pairs of points. These points are called nodes or vertices and the lines connecting the nodes are referred to as edges. A graph is denoted as  $G$  or  $G(V, E)$  where  $V$  represents the set of nodes and  $E \subseteq V \times V$  represents a set of edges of graph  $G$ .  $n$  is often used to represent the number of nodes  $|V|$ , and  $m$  to represent the number of edges (Gao *et al.*, 2009). The importance of representation of graph was initially introduced by Leonhard with a challenge being to traverse each bridge only once (Chartrand *et al.*, 2019). Euler denoted four land areas by vertices and seven bridges to represent the edges. Euler demonstrated that no more than 2 vertices can have number of edges of odd parity joining them to the other vertices of the graph for the existence of such a path. Number of edges of odd parity incident on all 4 vertices of the Konigsberg bridge graph implied that it is not possible to determine such a path. This discovery by Euler led to birth of a branch in mathematics called graph theory (Yegnanarayanan, 2020). In mathematics and particularly the field of graph theory, networks are usually referred to as graphs (from the Greek “*graphos*”, meaning something that is “*drawn*” or “*written*”). Graph theory then denotes the mathematical discipline that is concerned with the study of such structures and the modeling of relationships between objects (Gross, *et al.*, 2018).

### 2.7.1 Types of graphs

Graphs are classified into two types based on the presence or absence of the direction that links the nodes or elements. When there exists a directed link from one node to the other, the graph is referred to as a directed graph or diagraph while the graph with edges which are bidirectional is called undirected graph. For the undirected graph, the adjacency matrix is symmetric whereas the adjacency matrix for the diagraph is asymmetric. If there exist self-loops present in  $G(V, E)$ , then the total number of elements in  $E$ , which is denoted as  $|E|$  is a maximum of  $\frac{p(p-1)}{2}$  for undirected graphs and a maximum of  $p(p-1)$  for a diagraph where  $|V| = p$ . If set  $|E| = \frac{p(p-1)}{2}$  then  $G$ , is called a complete graph (Samanta, *et al.*, 2021).

#### 2.7.1.1 Undirected graph versus a directed graph

Undirected graph is defined as  $G = (V, E)$ , with  $V$  being a set of finite nodes features in a network, and  $E$  represents a finite set of edges that connects to a network node. The edge that connects these two nodes are usually undirected. A symmetric adjacency matrix  $A$  describes an undirected graph,  $G$ .  $A$  is a  $(|V| \times |V|)$  matrix, where  $a_{ij} = 1$  if and only if  $(i, j) \in E$ ; otherwise,  $a_{ij} = 0$ . Most of the biological networks, like protein- protein interaction networks do not consider direction of the action and for this reason, they are built using undirected graphs (Liu *et al.*, 2020). A directed graph is as defined as  $G = (V, E)$ , however the edges of the graph are directed. The direction means that the two nodes that are associated have an order of their relationship. For example, edge  $e = (i, j)$  is an ordered pair of nodes  $i$  and  $j$ , where  $i$  is the starting point of  $e$  and  $j$  is the end point of  $e$  (Liu, *et al.*, 2020). The meaning of this arrangement is that, for every edge of the graph, there exist a definite direction from the start to the end of the edge. These edges are used in describing occurrence of biological reactions. For example, when there is a relationship between a transcription factor and the regulated gene this becomes an orderly relationship which makes the regulatory network to be constructed as directed networks. Additionally, some computational methods can only be represented as a directed graph (Liu *et al.*, 2020).

### 2.7.2 Graph connection

A graph  $G(V, E)$  is regarded as disconnected if there is a pair  $u_r, u_s \in V$  with no path that exist between them. If there is a path of length 1 between  $u_r$  and  $u_j$  then it's called the edge that joins  $u_i$  and  $u_j$ . If there is a pair of vertices  $u_r, u_j$  and a path between them in in graph  $G$ , then in that it is a connected graph. A weakly connected graph is the one that doesn't have any path that connects two nodes. A matrix of a weakly connected graph is random whereas a graph is strongly connected if in each pair of nodes, there exist an edge connecting them (Figure 2.7).

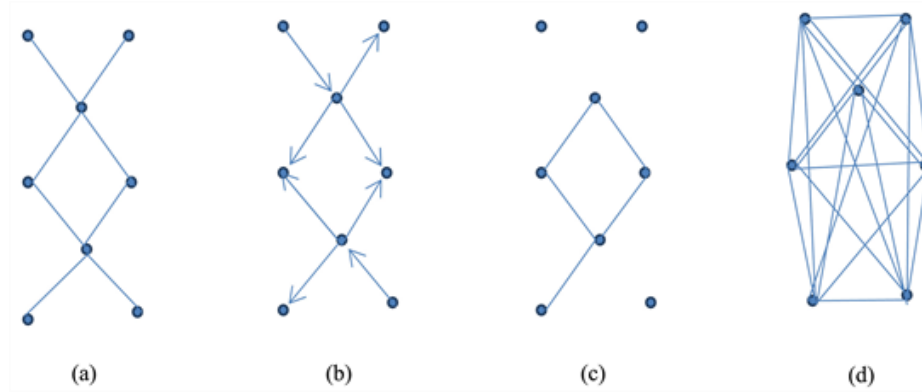


Figure 2.7: (a) Undirected graph; (b) Directed graph; (c) Undirected graph (disconnected) (d) complete undirected, graph (Yegnanarayanan, 2020).

### 2.7.3 Network topologies

The way in which nodes and edges are arranged within a network is referred to as network topology (Behzadi & Ranjbar, 2019). Topological properties can apply to the network as a whole or to individual nodes and edges (Chiang & Yang, 2004). Properties of graphs are important in unravelling useful information contained in a network. A crucial aspect of any network analysis is the ability to extract useful information that would have been difficult to discover if each component was to be examined individually. Therefore, network properties, especially topological properties, help identify relevant substructures within a network (Bansal *et al.*, 2018)

### 2.7.3.1 Global topology

Studies of the global topology of graphs shed light on their overall organization, such as (i) the overall connectivity of the network, (ii) the distribution of edges across vertices, (iii) the degree of clustering in the network, or (iv) the distribution of path lengths in the network. For instance, the overall connectivity of the network can be represented by the edge density

$$\rho = \frac{|E|}{\max |E|} \quad (2.16)$$

Where  $\rho = 1$  implies a complete graph, while a graph with  $\rho \ll 1$  means that the graph is sparse. The graph diameter is defined as the length of the longest path:  $D = \max_{i,j} d(i,j)$ , and the mean path length is defined as:

$$L = \sum_{i,j} \frac{d(i,j)}{\max |E|} \quad (2.17)$$

#### Small-world network

If path length denoted as  $L$  grows sufficiently slow, for example if  $L \propto \ln(N)$  the graph is said to represent a small-world network (Watts & Strogatz, 1998) (Figure 2.8). The small-world property implies that any target vertex  $v_b$  can be reached from a source vertex  $v_a$  by traversing only a small number of edges. Another global property of the graph is described by its degree distribution.

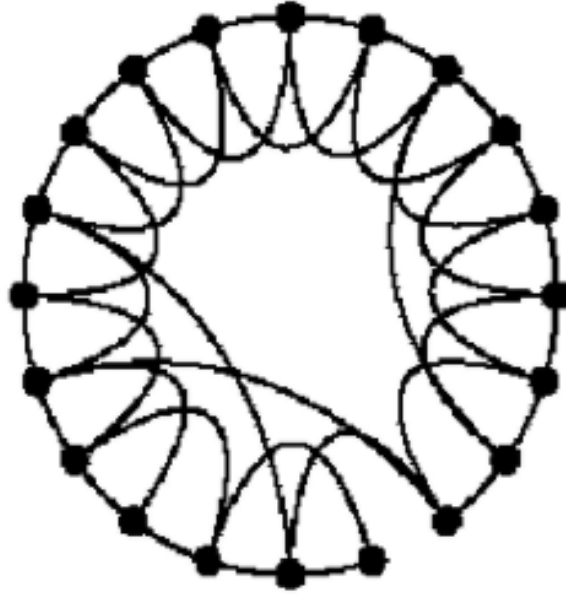


Figure 2.8: **Small-world network** adapted from Watts & Strogatz (1998).

Specifically, vertex  $v_j$  is a neighbor of vertex  $v_i$ , if  $e(i, j) \in E$ . The number of neighbors that a vertex has is referred to as the degree or degree centrality (DC) of that vertex. For an unweighted graph, the degree centrality of vertex  $v_i$  for an undirected network is computed as:

$$DC(v_i) = \sum_{j=1}^n a_{ij} \quad (2.18)$$

In directed networks, there is a distinction between in-degree  $DC_{in}$  since only incoming edges are counted.

$$DC_{in}(v_i) = \sum_{j=1}^n a_{ij}, DC_{out}(v_i) = \sum_{j=1}^n a_{ji} \quad (2.19)$$

If the probability  $p(DC(v_i) = k)$ , the probability that a vertex  $v_i$  in the graph exhibits a degree centrality  $DC=k$ , and can be modeled by a power-law distribution as

$$P(DC = k) \sim k^{-\gamma} . \quad (2.20)$$

### Scale free graphs

A graph is said to be scale free if most vertices in the network have very few incident edges while few vertices have a large number of incident edges. In random networks on the other hand, vertices tend to have similar degree values distributed around a mean degree  $\langle k \rangle$ . A random network was initially proposed by Gilbert and is denoted as  $G(n, p)$  constructed with  $n$  vertices  $V = \{v_1, v_2, \dots, v_n\}$  where each possible edge is included with probability  $p$  (Figure 2.9)



Figure 2.9: **Scale free graphs** (Whigham & Spencer, 2021).

Following the description provided by Barabasi (2013) in such a random graph, the distribution of degree centralities can instead be modeled by a binomial distribution:

$$P(DC = k) = \binom{n-1}{k} p^k (1-p)^{n-1-k} \quad (2.21)$$

In the case where  $\langle k \rangle \ll n$  by the Poisson distribution

$$P(DC = k) = e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!} \quad (2.22)$$

### 2.7.3.2 Local topology

Local topological properties of the network allow the identification of substructures or vertices with characteristics. Several metrics have been developed to prioritize vertices in terms of their connectivity pattern or other related measures of centrality within networks. In addition to degree centrality, local topological measures allow identification of bottlenecks e.g., vertices that connect different network modules (Yu *et al.*, 2007) due to a high betweenness centrality (BC) of these vertices. Specifically, the BC for a vertex  $v_i$  is formally defined by:

$$BC(v_i) = \sum_{s \neq i \neq t} \frac{\sigma_{st}(v_i)}{\sigma_{st}} \quad (2.23)$$

where  $\sigma_{st}$  counts the number of shortest paths from vertex  $v_s$  to vertex  $v_t$  and  $\sigma_{st}(v_i)$  is the representation of the numbers of shortest paths originating from vertex  $v_s$  to vertex  $v_t$  that also include vertex  $v_i$ .

Local clustering coefficient is another example of local topological metric used to identify vertices that are linked to highly connected clusters in the network. Let  $N(v_i)$  be the set of vertices that are neighbors of vertex  $v_i$ . Then  $CC(v_i)$  can simply be defined as the fraction of actual versus possible connections between all the pairs of such neighbors (watts, 1997). In a directed network, a total of  $|N(v_i)|(|N(v_i)| - 1)$  nodes are present. Let

$m_i$  denote the number of observed connections between the neighbors of vertex  $v_i$  the CC of the vertex  $v_i$  in a directed network is defined as:

$$CC(v_i) = \frac{m_i}{|N(v_i)|(|N(v_i)|-1)} \quad (2.24)$$

## 2.8 Graph-based feature selection methods

To uncover similarity associations from data, graph-based algorithms have recently been applied in machine learning techniques. Graph-based approaches for feature selection give an underlying manifold structure as a universal foundation for reflecting feature relationships. Graph-based approaches have been used to handle feature selection difficulties in several studies. For example, a dense subgraph discovery strategy is used to solve the unsupervised feature selection problem in (Yan *et al.*, 2021). In, another feature subset selection approach based on clustering is described for high-dimensional data (Moslehi & Haeri, 2021). For similar grouping features, this study used a graph-theoretic clustering technique. A hypergraph-based technique for feature selection was proposed by Zhou *et al.*, (2022). This study considered the related class label of each sample when evaluating the applicability of distinct features using an information-theoretic criterion. The notion of graph clustering using node centrality measure is merged with the unsupervised feature selection process in (Moradi & Rostami, 2015). The authors Ghaemi & Feizi-Derakhshi (2016) expanded on this study by selecting features that were more informative. In order to rank features based on their importance, Henni *et al.*, (2018) employed Google's PageRank centrality measure.

Hashemi *et al.* (2020) developed another graph-based feature selection method for multi-label high-dimensional dataset. A PageRank centrality measure was used by the authors of this study to rank the properties based on their importance in the graph. In addition,



correlation distance criteria was used in their study to eliminate redundant features. Li *et al.*, (2019) suggested an unsupervised graph-based feature selection method for high-dimensional data. Laplacian graph and local geometrical structure were employed in this study to better depict features space. By conducting feature selection and subspace learning in the sample self-representation framework, Zhu *et al.*, (2017) proposed a subspace clustering guided unsupervised feature selection strategy in which a transformation matrix was projected to the original data to their low-dimensional space by conducting feature selection and subspace learning. They then created a dynamic and intrinsic affinity matrix using the rank constraint and the affinity matrix generated directly from the source data. Final clustering results were determined using the affinity matrix learned from the low-dimensional space. (Zhu *et al.*, 2017).

Other methodologies have been presented together with machine learning approaches to tackle the problem of inferring meaningful biological networks. One of the earliest (but still widely used) proposed approaches is based on the “guilt-by-association” principle. This implies that when two genes show similar expression profiles, the assumption is that they are related biologically by either direct or indirect interaction. Networks generated from biological data represent a tool to investigate complex biological systems (Yu *et al.*, 2015). “Guilt-by-proximity” implies that genes that lie closer to each other on the network are more likely to lead to the same phenotype (Figure 2.10).

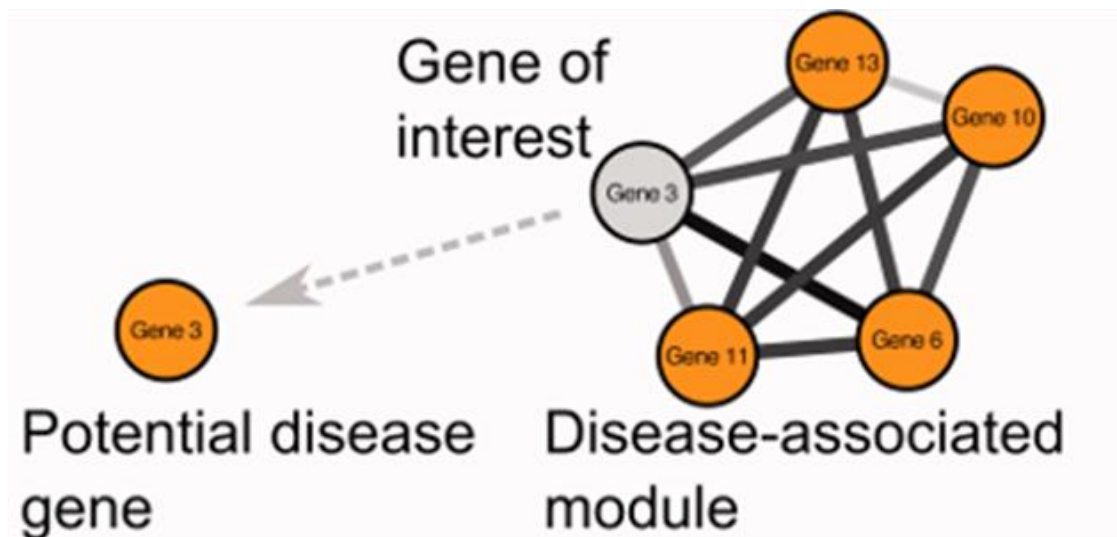


Figure 2.10: **disease association module** (van Dam *et al.*, 2017)

Zhang and Zeng (2019) increased disease gene prediction model performance by combining disease phenotype, biological function, and network topology similarity. Lei *et al.*, (2019) combined RWR and Pearson Correlation Coefficient (PCC) to measure similarity of two proteins. Zhang *et al.*, (2014) proposed a method, named ESFSC, based on RWR to rank disease genes. The innovation of ESFSC is enlarging seed nodes with known disease genes and their k-nearest neighbor nodes. Mamoshina *et al.* (2018) used publicly available gene expression profiles of young and aged tissue from healthy donors to conduct their research. Differential gene expression and pathway analysis were used to compare profiles of young and elderly muscle tissue, as well as data preprocessing for a set of machine learning methods. Based on blood gene expression profiles, Kaletsky *et al.*, (2019) conducted a study to establish a transcriptome signature that may be used to diagnose people with autism spectrum disorder (ASD) compared to controls. Ganegoda *et al.*, (2015) introduced a novel method called proximity disease similarity algorithm (ProSim), which prioritizes disease genes by considering both aforementioned qualities

### **2.8.1 Graph-based filtering metrics**

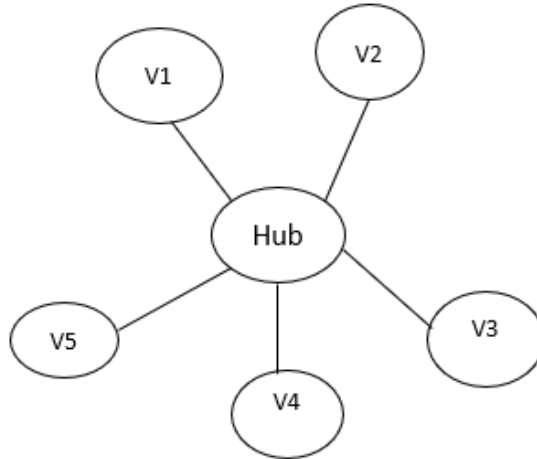
Features that are associated with each other tend to have similar functions based on the “guilt by association” assumption. In the case of biological data, many forms of networks that characterize relationships among features or genes have been employed in disease gene prediction. Graph-based analysis algorithms have been categorized into classes such as centrality-based methods, distance-based methods, random walk-based methods (Luo *et al.*, 2021).

#### **2.8.1.1 Network Centralities and Node Ranking**

Nodes in a network can be sorted or ranked based on their unique properties and depending on the research question at hand. Central nodes or other intermediate nodes are crucial in affecting the topology of the network for example finding nodes that tend to interact with many other features/ proteins or finding molecules that play important roles in genes expression stimulation (Singh *et al.*, 2022).

#### **2.8.1.2 Degree Centrality**

Degree in a network gives an important node that has the highest number of interactions with other nodes. The degree of node  $i$ , is usually calculated as  $C_{d(i)} = deg(i)$  for undirected graph and in case of a directed graph, every node is defined by two-degree centralities. These are  $C_{d in}(i) = deg in(i)$  and  $C_{d out}(i) = deg out(i)$ . Nodes that have very high degree are regarded as hub because they are connected to many neighbors (Figure 2.10). If such nodes are removed, then the topology of the network is highly affected such that the network becomes highly disconnected. Biological networks have been shown to be very robust against any random disconnection. However, if hub nodes are disrupted, then this can lead to a failure of the system (Jardim *et al.*, 2019).



**Figure 2.11:** Hub nodes in a network

### 2.8.1.3 Closeness Centrality

This centrality measure designates the most important nodes which communicate faster with other neighboring nodes within a network. With  $G = (V, E)$  being an undirected graph, the centrality measure is defined as:

$$C_{clo} = \sum_{t \in v} \frac{1}{|V|} dist(i, j) \quad (2.25)$$

where  $dist(i, j)$  denotes the shortest path  $p$  between the two nodes  $i$  and  $j$ . Closeness centrality has also been used in identification of non-random structure of genomic as well as proteomic networks (Halder *et al.*, 2020). Any reduction in closeness centrality measure of the components in the network has a consequence in increasing the distance between pathways throughout the entire network (Agrawal *et al.*, 2018). In biological networks, closeness centrality has been shown as one of the best centrality measures in identifying critical nodes in a network (Liu *et al.*, 2020)

#### 2.8.1.4 Betweenness Centrality

This measure shows nodes that are in-between neighboring nodes, and their rank is usually higher. These nodes play a crucial role in facilitating the communication between two neighbors. Therefore, betweenness centrality indicates those crucial nodes that lie between the paths and other nodes in a network (Feng *et al.*, 2021). For individual nodes  $i, j, w \in V(G)$ , with  $\sigma_{ij}$  being all the shortest paths that lie between  $i$  and  $j$  and  $\sigma_{ij}(w)$  which is the total number of shortest paths from  $i$  to  $j$  that passes through  $w$ . Moreover, for  $w \in V(G)$ , let  $V(i)$  is the set of all pairs of ordered nodes,  $(i, j)$  in  $V(G) \times V(G)$  such that  $i, j, w$  are all separate.

#### 2.8.1.5 Eigenvector Centrality

This measure ranks the higher nodes which have been connected to neighbors that are important. if  $G = (V, E)$  being undirected graph and  $A$  is its network  $G$  adjacency matrix then the eigenvector centrality becomes eigenvector  $C_{eiv}$  of the main eigenvalue  $\lambda_{max}$  in absolute value in that  $\lambda C_{eiv} = AC_{eiv}$ . If  $A$  is the adjacency matrix of a network  $G$  with  $V(G) = \{v_1, \dots, v_n\}$ , and,  $\rho(A) = \max_{\lambda \in \sigma(A)} \lambda$  then the eigenvector centrality  $C_{eiv}(v_i)$  of the node  $V_i$  is given by the  $i^{th}$  coordinate  $x_i$  of any normalized eigenvector which satisfies the condition  $Ax = \rho(A)x$ . These algorithms have been used for efficient page ranking on the web. Another wide application of this measure has been used in identification of genetic interactions in cancer studies (Henkel *et al.*, 2019), gene-disease associations (Hwang *et al.*, 2019) or network hubs in PPI networks (Amala & Emerson, 2019).

#### 2.8.1.6 Eccentricity Centrality

This measure shows how easy a node is accessible from other nodes. If  $G = (V, E)$  is an undirected graph, then eccentricity is usually calculated as  $C_{ecc} = \frac{1}{\max\{dist(i,j)\}}$  with

$dist(i, j)$  being the shortest path connecting two nodes  $i$  and  $j$ . The eccentricity  $C_{ecc}$  of a vertex  $V$  is the longest distance between  $v$  and any other adjacent vertex (Dragan, 2020). These centrality measures such node degree, closeness, betweenness, and eigenvector have been very useful in the road networks to identify traffic congestion (Jayaweera *et al.* 2017). These measures have been applied in analyzing urban road transport in study done by Wang *et al.*, (2017). Another earlier study that used centrality measure is by Grunspan *et al.*, (2014) where they studied the interaction among students of a class from a student network. They did a social network analysis and concluded that there was improved character, knowledge, and relationship among students. The degree and betweenness centrality measures were used in analysis of relationship among students and the authors found out that there was a relationship between interaction network student's performance. Centrality measures plays a crucial role in network analysis however there should be proper selection of the measure based on the application (Das *et al.*, 2018).

### **2.8.1.7 Distance-based methods**

Distance based feature selection is a representation of the distance between features and the targeted feature set within a metric space (Tan *et al.*, 2020). Distance-based feature selection FS is classified into two categories which are based on the output which is generated. These categories are either as a subset of features or a ranked list of partial or complete features. (Bolón-Canedo, *et al.*, 2016). Methods that provides rated list of features are also referred to as rankers and they are the most used subcategory in filtering methods. Rankers rely on a given evaluation measure like information dependency or distance, that enable measuring and sorting of features based on their predictive importance. Evaluation of features is done independently on every feature in these measures. The most common distance measures that are used in feature selection are Euclidean distance to the complex distances such as Minkowski distance (García, *et al.*, 2015).

Liu & Zhang, (2016) developed three unsupervised feature selection methods that were based on the effective distance. These features methods were effective distance-based Laplacian Score and two effective distance-based Sparsity Scores 1 and 2. They demonstrated using experimental results that their new distance-based feature selection methods could achieve much better performance as compared with conventional methods that are developed using Euclidean distance. The authors recommended the use of the proposed approaches in dimensionality reduction and in graph-based learning algorithms (Liu & Zhang, 2016). A fault diagnosis scheme that was derived from envelope analysis based on the Euclidean distance as suggested by Li *et al.*, (2014). This algorithm could detect faults using an intelligent system even with bearings being under different default levels. (Li *et al.*, 2014). Another study on nonlinear feature building technique that was derived from Euclidean distance was presented by (Feng *et al.*, 2017) that could point out nonlinear features from difference filters (DIFs). In their study, ED was applied between DIFs, however feature reduction as well as feature ranking based on Euclidean distance to improve on performance of classifiers were unknown.

Patel & Upadhyay, (2020) presented another feature ordering and selection approach which was called Feature Ranking and Subset Selection based on Euclidean distance (FRSSED). They considered two bearing databases for verification of how robust the approach was. Authors used feature extraction on the selected IMF using several statistical measures. This was followed by introduction of the extracted features into the proposed FRSSED algorithm for ordering. Ordered features were then classified using different classifier and the resulting accuracy as well as the computation time compared. The authors reported improvement in accuracy and computation time for the proposed approach with reduced feature subset (Patel & Upadhyay, 2020).

Shahee & Ananthakumar, (2020) proposed a feature selection approach that was based on distance. Known as ED-Relief that was meant to use distance measure to handle simultaneous occurrences of within and between class imbalances. They tested the method using both simulated and real-life datasets and compared the results with the well-known distance based such as effective distance-based Laplacian Score (EDLS), and two EDSS-1 and EDSS-2. The authors claimed enhanced performance based on accuracy metrics, revealing that ED-Relief performs better or comparable to other accuracy measures. As a result, the ED-Relief distance measure is highly good in incorporating simultaneous imbalance for identifying features that better distinguish across the classes (Shahee & Ananthakumar, 2020)

Fu *et al.*, (2020) proposed an algorithm known as sssHD that was based on the Hellinger distance (HD) combined with sparse regularization techniques. The authors reported the sssHD generality since it could combine different re-balance samplings like under-sampling and over-sampling, could change the sparse regularization structure based on the characteristic of the predictor matrix, like LASSO (Yuan *et al.*, 2017). If the predictors could possess some form of group structure and lastly, if necessary, SVM classifier used in sssHD could be replaced such as discriminant analysis, Naïve Bayes, random forest etc. (Fu *et al.*, 2020). In disease genes prediction, distance-based were the first to be developed. The length of the shortest path (distance) in networks was used in these ways to see if a healthy gene was linked to a disease gene. Unknown genes which have not been given a function can also be predicted to be associated with diseased genes as long as the distance is smaller than the set threshold.

Banka & Dara, (2015) introduced a Hamming distance as a proximity measure used to update the velocity of particle(s) in binary PSO framework for selecting only the important subset of feature. They did their experiment on three colon cancer datasets and evaluation was done using classification accuracy. This revealed the importance of proposer selection of the preprocessing method and concluded that HDBPSO combined with Hamming



distance as a proximity measure can find relevant features from gene expression data with better performance (Banka & Dara, 2015). Cheng *et al.*, (2018) constructed a network to determine the relationship between a drug and a given disease. The expected distance between group of proteins were calculated in a network and z-score was calculated by conversion of non-Euclidean distance to a normalized distance. Four network-predicted associations were used to test relationship using large healthcare databases with more than 220 million patients. In conclusion, the authors demonstrated that drug repurposing can be facilitated by a unique combination of protein-protein interaction network closeness and large-scale patient-level longitudinal data supplemented by mechanistic in vitro research (Cheng *et al.*, 2018). In Banuchitra, (2021), the Gaussian kernel was used to construct similarity scores using a distance-based approach, and unknown genes with greater similarity scores were expected to be disease-associated. Therefore, distance-related approaches are still valuable, and they have been used in extracting features in conjunction with many machines learning-based methods (Luo *et al.*, 2021).

### 2.8.1.8 Random walk-based methods

The significance of two neighboring nodes in a network is captured by proximity based on node-to-node, and it is an important study subject in data mining (Shin *et al.*, 2021) Due to its capacity to evaluate both the local and global structure of the graph, Random Walk with Restart (RWR) is an extensively used proximity metric (Lin *et al.*, 2020). In a network, the Random Walk algorithm imitates a walker by travelling from a current node to a randomly chosen next node or by walking back to the source nodes with a back-probability of (0,1). (Le & Pham, 2017).Random walk with restart is defined as  $G(V, E)$  for weighted graph which has a set of nodes  $V = \{v_1, v_2, \dots, v_N\}$  and a set of links  $E = \{(v_i, v_j) | v_i, v_j \in V\}$ , a set of source/seed nodes  $S \subseteq V$  and a  $N \times N$  are regarded as adjacency matrix  $W$  of link weights. RWR is described as follows:

$$p^{t+1} = (1 - \gamma)W^t p^t + \gamma p^0 \quad (2.26)$$

Where  $p^t$  is a  $N \times 1$  is the probability vector of  $|V|$  nodes at a time in step  $t$  of with the  $i$ th element representing the probability that the walker is at node  $v_i \in V$  and  $p^0$  is the  $N \times 1$  is the initial probability vector which is defined as:

$$p^0 \begin{cases} \frac{1}{|S|} & \text{if } v_i \in S \\ 0 & \text{Otherwise} \end{cases} \quad (2.27)$$

$W'$  denotes the transition of the matrix of the graph  $(i, j)$  being element of  $W'$ . The probability that the walker at  $v_i$  moves to  $v_j$  among  $V \setminus \{v_i\}$ , Formally,  $(W')'_{ij}$  is given as:

$$W'_{ij} = \frac{(W)_{ij}}{\sum_j (W)_{ij}} \quad (2.28)$$

Random walk with restart has been applied in several areas of research such as image retrieval (Yang *et al.*, 2020). The authors provided a random walk model with nodes denoting images and the weights of edges representing image similarities. The images that were labelled as relevant and non-relevant by the users were considered as seed nodes that solved the random walker problem as well as the ranking score at every unlabeled image. The probability that a random walker would start from an image and reach a relevant seed without encountering a non-relevant image on the graph was calculated by considering characteristic of and spatial relations of images image when a random walk was conducted. To realize this, feature weighting using Laplacian Score together with K-nearest neighbor (KNN) that connects to the random walk were considered (Wang *et al.*, 2019). Another study that used random walk with restart was by Wu *et al.*, (2019) who developed a random walk-based image registration technique that could examine the solution search space with efficiency. Their method employed available information of probabilistic solution therefore reducing computation cost (Wu *et al.*, 2019). Chang & Wang, (2018) presented a novel framework that was based on the sub-Markov random walk for interactive image segmentation. The new auxiliary nodes made their framework more flexible that could solve a problem associated with thin and elongated parts (Chang & Wang, 2018). Earlier study by Yildirim & Coscia, (2014) defined a new similarity

measure that utilized a practical procedure in extraction of unipartite graphs without having *a priori* assumptions on underlying distributions. This measure captured relationship among objects using a likelihood that a random walker could make a sequential pass through the of the bipartite graph (Yildirim & Coscia, 2014).

Another study by Xia *et al.*, (2016) proposed a method known as CARE that incorporated the relationship between authors historical preferences for scientific article recommendation. The assumption here was that some researchers tend to search for articles that have been published by the same. Therefore, authors construct a graph based on the relationship of co-authors' information employing a random walk with restart that generates a possible recommendation list (Xia *et al.*, 2016). Random Walk-based approaches have been used in research to predict disease genes prediction. Random walk-based algorithms iteratively transmit prior knowledge from each node to surrounding nodes for a predetermined number of steps or until convergence. The final value of a node is basically influenced by its direct neighbor's values, which in turn affects their neighbors (Luo *et al.*, 2021). In earlier study by Köhler *et al.*, 2008, they proposed the first RWR algorithm for diseased gene. Jiang *et al.*, (2015) applied RWR to three disease similarity networks and nine gene similarity networks, then combined all the findings in prioritizing disease genes using a weighted Fisher's technique (Jiang, 2015). Valdeolivas *et al.*, (2019) constructed heterogeneous networks with the same nodes connected with each other allowing the transition of the random walk between different networks, which authors reported improvement in accuracy of the prediction (Valdeolivas *et al.*, 2019). Lei & Bian, (2020) proposed a method called RWRKNN, which integrated RWR and k-nearest neighbors (KNN) in predicting possible associations between circRNAs and diseases. The proposed algorithm used weighting features together with global network topology information to classify features using KNN which provided prediction scores of each paired circRNA-disease (Lei & Bian, 2020).

Vural *et al.*, (2019) developed a computation model that used similarity matrices for circRNA and disease respectively through application of Gaussian followed by random walk with restart algorithm that was applied on the combined matrices (Vural *et al.*, 2019).

Wang *et al.*, (2019) provided a disease prediction model that was based on internal inclined random walk with restart (IIRWR) to infer potential lncRNA-disease associations. The approach introduced a concept of clique that made the process of the random walk to have an internal tendency. Wang *et al.*, (2019) and Li *et al.*, (2017) used RWR algorithm to search for novel genes by use of known genes as seed nodes. To enhance the reliability of the model, screening, permutations test and interaction test were used to select important genes that were obtained from RWR algorithm (Li *et al.*, 2017).

## 2.9 Graph clustering techniques

Cluster analysis involves gathering similar data points in the same group such that all data points in one group have similar traits to each other than the data points in other groups. This technique is an unsupervised used mostly in exploratory data analysis. In classification and regression models' data sets with tagged class labels are used unlike clustering where the data sets have no class labels are provided the concept of similarity degree is the key to cluster analysis because clustering results depends on the similarity measure adopted. In this section we review some of the most famous and most applied algorithms in analysis of biological data.

### 2.9.1 Partitioning clustering

Partitioning clustering works by obtaining a section of data in which every point belongs to a unique cluster. A K-Means Clustering Algorithm is an example of the best well-known algorithm for K-Means clustering,  $X = \{x_1, \dots, x_n\}$  is defined a set of  $N$  points in a multi-dimensional space and  $K$  is the integer value, then the algorithm finds a set of  $K$  vectors  $\mu_k$  that minimize the Within Cluster Sum of Squares (WCSS):

$$WCSS = \sum_{n=1}^k \sum_{x_i \in C_h} d(x_i, \mu_h) \quad (2.29)$$

where  $C_h$  is the  $h$ -th cluster and  $\mu_h$  is the corresponding centroid. K-means can solve most of the practical problems whose results are clusters with hyper spherical shape, and it also runs in approximately linear form. Its main drawback is the possibility of being easily

trapped in local minima at the phase of optimization process. It is also sensitive when starting initialization of the centroids (Serra *et al.*, 2018). A random seed is usually set before its execution to ensure reproducibility of the experiments. K-means is also sensitive to noise since they use the centroids means. Another challenge associated with k-means is that the number of clusters required are fixed even if it's unknown in the data it must be estimated using cluster analysis. An algorithm that uses partitions around medoids using the medians that are centrally located object within the cluster as medoids were introduced to address the issue of the outlier effects on the cluster prototypes (Xu & Wunsch, 2009). The improved k-medoids has been used in areas of study such as calculating the optimal medoids among the sensor nodes (Wang *et al.*, 2018), Items rating probability distribution (Deng *et al.*, 2019), epilepsy signal detection (Zhang *et al.*, 2021) and clustering of microarray data (Bustamam *et al.*, 2018).

### **2.9.2 Hierarchical clustering**

Hierarchical clustering algorithms have been the most preferred clustering methods that has been commonly used in identification of bioinformatics structures (Reddy & Vinzamuri, 2018). Their resulting hierarchical tree known as dendrogram that represents a nested set of partitions is produced. This dendrogram is usually cut at a particular level to produce a partition of  $K$  disjoint clusters. Squared Euclidean distance between cluster centroids is used to calculate the centroid linkage. Data can be represented in a Euclidean space, according to the assumption. Hierarchical clustering has been frequently used in air pollution analysis (Govender & Sivakumar, 2020), multihop Internet of vehicles communication (Dutta *et al.*, 2020) and in bioinformatics (Xiong *et al.*, 2018; Gobin *et al.*, 2019; Yousef *et al.*, 2021; Waylen *et al.*, 2020; Teng *et al.*, 2022).

### **2.9.3 Density-based clustering**

Clusters in this sort of clustering technique are assumed to be dense groups of points in the data space separated by lesser density regions. The DBSCAN algorithm initially proposed by Ester *et al.*, in (1996) is the most well-known clustering algorithm based on

density. A density-reachability model uses two parameters to connect sites within a certain distance: ( $\epsilon$ ) distance threshold and minPts (minimum number of items in a cluster). The algorithm starts by identifying the initial neighbors of every point with a distance less than minPts, then moves on to identifying core points with more than minPts neighbors. It then ignores the noncore points and looks for the core points' related components on the neighbor graph. If the cluster is a  $\epsilon$ -neighbor, each noncore point is finally assigned to it; otherwise, the point is considered noise. DBSCAN does not require a set number of clusters to be returned because it can detect clusters of various shapes and is resistant to outliers. This clustering approach has been applied in the integration of environmental data for desertification (Peng *et al.*, 2021), picture clustering (Ren *et al.*, 2020), and bioinformatics (Thrun & Ultsch, 2021; Mandal & Sarmah, 2018; Mallik & Zhao, 2020).

#### **2.9.4 Spectral clustering**

Spectral clustering techniques use the similarity matrix's eigenvalues and eigenvectors to conduct dimensionality reduction before clustering items in a lower-dimensional space. When creating the affinity matrix, most spectral clustering algorithms consider sample correlation. The clustering problem is recast as a problem of finding the optimum graph partitioning since the affinity matrix may be thought of as a graph. This modification can significantly reduce clustering complexity and, as a result, play an important role in spectrum clustering. When computing the similarity between two samples, sample self-representation assumes that each sample is represented by other samples on the same subspace (Hu *et al.*, 2017; Zhu *et al.*, 2018). Spectral clustering has been used in a variety of applications, including social network clustering (Li *et al.*, 2018), image processing (Chen *et al.*, 2017; Cribben & Yu, 2017), and bioinformatics (Hobbs *et al.*, 2017).

#### **2.9.5 Affinity propagation**

Message parsing idea is used between the data points in the affinity propagation clustering technique. The most representative items are found, and clusters are constructed around them. Its input is a pairwise similarity between data points. It works by treating several

data points as candidate exemplars and sending signals back and forth between them until the best exemplars and clusters emerge. There are two sorts of messages used in the message passing procedure. The evidence gathered on how suitable point  $k$  is to serve as an instance of point  $I$  is conveyed from data point  $I$  to candidate instance point  $k$ , considering other possible examples of point  $i$ . The availability message sent from candidate example point  $k$  to point  $I$  representing the evidence obtained on how appropriate it would be for point  $I$  to choose point  $k$  as its example, given support from other points for which point  $k$  may serve as an exemplar (Serra *et al.*, 2018). The affinity propagation method does not require several clusters to be chosen, and it generates more clusters with unequal cluster sizes than other clustering methods. Even though the number of groups isn't required as an input, affinity propagation necessitates the establishment of a parameter (preferences) for each point: points with higher preference values are more likely to be chosen as samples. Unless otherwise stated, the total number of clusters is usually determined by the input preference values, which are initially set to the median of the proximities entered. This strategy has been used to solve difficulties in computational biology (Fonseca *et al.*, 2017; Busch *et al.*, 2020).

### **2.9.6 Projective clustering**

Projective clustering is used to locate subsets of input elements in subspaces of the original space that are strongly linked. The purpose is to locate those subsets of input components that are significantly associated in subspaces of the original space using high-dimensional data (Yu *et al.*, 2017). A subset of correlated points and their corresponding subspace is referred to as a projective group. When projected into the related subspace, all the points in the group are near together, yet they can be scattered over a full-dimensional space. Projective clustering methods are very effective for extracting or indexing data sets where full-dimensional clustering is insufficient (as is the case for most high-dimensional data sets). Furthermore, unlike global dimensionality reduction, these techniques produce projective groups that exist in distinct subspaces, making them more general. A set of

projective clustering methods were used in clustering cancer gene expression datasets to obtain a more stable and robust solutions to noise (Yu *et al.*, 2017).

## 2.10 Graph Similarity measures

A similarity score between gene pairs is calculated using a variety of methods, each with its own set of advantages and disadvantages. In terms of detecting gene correlations and performance on huge data sets, simple Pearson or Spearman correlation is frequently utilized and outperforms more complicated approaches (Dutta *et al.*, 2018). Although Pearson requires a linear correlation, normally distributed numbers, and is sensitive to outliers, it is the most widely used correlation measure. The rank correlation of Spearman is more robust, but it is also less powerful. Mutual Information (MI) is another commonly used metric for describing non-linear gene relationships (Liu *et al.*, 2021).

### 2.10.1 Pearson's correlation coefficient (PCC)

Pearson's correlation is a bivariate normal distribution-based measure of the linear relationship between two continuous random variables. Data is said to be near bivariate normal distribution only when the sample size is large enough. Regardless of whether the joint distribution of two random items is normal or not, the Pearson correlation coefficient is extremely instructive regarding the degree of linear dependency between them (Hou *et al.*, 2022). If the data are normal, the Pearson correlation coefficient provides a precise and full representation of the relationship, and it may have considerable advantages for continuous data with no clear outliers. (Ovens *et al.*, 2021). This attribute makes it most employed metric for inferring the *co-expression* relationships in gene expression network. Pearson's correlation coefficient, which for a pair of genes  $g_i$  and  $g_j$  can be estimated from the expression. The Pearson correlation coefficient is defined as

$$R(i) = \frac{\text{cov}(x_i, Y)}{\sqrt{\text{var}(x_i)\text{var}(Y)}} \quad (2.30)$$



with *cov* being covariance and *var* being the variance. The estimate of  $r(x_i, x_j)$  is given by

$$r(x_i, x_j) = \frac{\sum_{k=1}^m (x_{k_i} - \bar{x}_i)(y_{k_j} - \bar{y}_j)}{\sqrt{\sum_{k=1}^m (x_{k_i} - \bar{x}_i)^2 \sum_{k=1}^m (y_{k_j} - \bar{y}_j)^2}} \quad (2.31)$$

where  $m$  is the number of samples  $x_i$  is the vector holding all  $m$  expression values of gene  $g_i$ ,  $x_{k_i}$  denotes the expression of gene  $g_i$  in sample  $s$  and  $\bar{x}_i$  is the mean expression of gene  $g_i$  across all samples. However, similarity scores can also be derived through other correlation measures such as Spearman's rank correlation coefficient or other gene association metrics (Shekhovtsov & Saġabun, 2020).). After the initial computation of association values, it is then possible to create a sparse network by choosing a '*hard threshold*' (Feng *et al.*, 2020, Zhang & Horvath, 2005) and setting edges between any pair of genes whose correlation value exceeds this threshold, or to generate a weighted network by the use of '*soft thresholding*', e.g., in the form of raising the absolute correlation value to a power  $\beta$ .

Mu *et al.*, (2018) in their study, tried to confirm the best splitting attributes and splitting points in the evolution of decision trees, PCC was used as a new measure of feature quality. The suggested solution parallelizes every component of the decision tree learning process, which comprises primarily of a parallel PCC-based splitting rule and a parallel splitting data strategy, using Map-Reduce technology. The experiment was carried out on a variety of UCI benchmark data sets with various scales. Based on results from 17 data sets, the authors reported that their approach outperforms numerous classic decision tree classifiers, such as BFT, C4.5, LAD, SC, and NBT in terms of computational resources required and classification accuracy. Other techniques used in co-expression measure are:

### 2.10.2 Mutual Information

Mutual information  $I(X;Y)$  is the degree of uncertainty in  $X$  due to knowledge of  $Y$  in probability and information theory and it is defined as below:

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \quad (2.32)$$

where  $p(x,y)$  is the joint probability that shows distribution functions of  $X$  and  $Y$ , and  $p(x)$  and  $p(y)$  being the marginal probability distribution functions for  $X$  and  $Y$ . Therefore, we can as well say  $I(X;Y) = H(X) - H(X|Y)$  where  $H(X)$  is the marginal entropy,  $H(X|Y)$  is the conditional entropy, and  $H(X;Y)$  is the joint entropy of  $X$  and  $Y$ . If  $H(X)$  represents the measure of uncertainty about a random variable, then  $H(X|Y)$  measures what  $Y$  does not say about  $X$ . This is the degree of uncertainty in  $X$  after knowing  $Y$ , confirming the intuitive sense of mutual information as the amount of information provided by knowing one variable about the other. A mutual information measure is employed in our method to quantify the information gain between features and between feature and class characteristics (Hughes *et al.*, 2020).

In gene expression data analysis, this method assigns a significant value (p-value) to each MI value based on permutation analysis as a function of sample size (Lall *et al.*, 2021). A technique that uses MI is mRMR, a multivariate filter that selects features with the highest relevance to the target class while also being minimally redundant, i.e., selecting characteristics that are maximally unlike each other. Instead of using MI to assess relevance and irrelevance of features it is easier to estimate from data since it has several desirable qualities of a measure of dependency, such as being bounded. If two random variables have a monotonic connection, it achieves its maximum selection. The addition of a free parameter ( $\lambda$ ) that determines the relative importance given to relevance and redundancy is one of its contributions. It should be noted that some of the offered filter approaches are univariate in terms of computing cost. When compared to other feature selection procedures, this means that each feature is analyzed independently, ignoring

feature relationships, which could lead to poor classification results (Bolón-Canedo, 2014).

### 2.10.3 Spearman's rank correlation coefficient

This is a non-parametric estimator approach does not rely on making assumptions regarding distributions of X and Y, and makes an estimation of monotonic association between the variables and is computed as:

$$sCor(X, Y) = 1 - \frac{6\sum_{i=1}^m d_i^2}{m(m^2-1)} \quad (2.33)$$

where,  $d_i$  represents the difference between the ranks of  $x_i$  and  $y_i$ .

The effectiveness of feature reduction and classification accuracy are linked to the relationships that exist between attributes and classes. This link is comparable in terms of properties. A correlation coefficient is therefore a measure that is used to calculate the relationship between qualities in general. Bivariate normal distribution, chi-square test for independence, rank correlation coefficient, and so on are some of the most common correlation coefficient measures. Spearman's rank correlation coefficient is a nonparametric measure of rank correlation that is statistical dependence between the ranking of two variables. It determines how effectively a relationship between two variables expressed by a monotonic function is. Although not widely used in coexpression networks, this measure has been applied in microarray classification (Xu *et al.*, 2018) and gene network (Quintana *et al.*, 2019; Hou *et al.*, 2019)

Multiple repetitions of experiments, especially in high-throughput data generation are performed on the same set of samples to investigate different aspects of the same phenomena such as gene expression, miRNA expression, etc. Multiple perspectives can be used to provide a better understanding of the fundamental principles of complex systems. Many Multiview clustering techniques have been devised to gain a better understanding of these complicated processes by integrating diverse viewpoints of the

data. Matrix factorization-based approaches for integrating clustering solutions acquired for each single view are some examples (Zong *et al.*, 2017). Other methods rely on tweaks to the standard k-means clustering technique. Other methods focus on the integrative analysis of networks constructed on each view, employing iterative optimization analysis based on local neighborhood, and finally applying spectral clustering to the final integrated matrix (Wang *et al.*, 2014).

### **2.11 Data discretization**

Discretization is a technique for reducing dimensionality that has been employed in big data analytics. This method converts continuous data into discrete data, which may then be utilized to develop machine learning models. Despite its origins in computer science and statistics, this technique has been embraced as a preprocessing step in biological data analysis (Gallo *et al.*, 2016). Discretization facilitates the use of methods for the inference of biological knowledge that require discrete data as an input by mapping real data into a generally limited number of finite values (Alagukumar & Lawrance, 2015). supervised and unsupervised data discretization techniques are similarly divided into two types. The unsupervised discretization works without relying on any class label information provided by the user to compute the discrete states of data whereas supervised methods put into consideration prior knowledge of data before performing the discretization (Gallo *et al.*, 2016).

This approach is widely used as a preprocessing step in biological data analysis. This is because the RNAseq data is continuous and had to be converted to discrete. By mapping real data into a small number of finite values, the purpose of gene expression data (GED) discretization is to make it possible to apply methods for biological knowledge inference that require discrete data as input. The biological problems that discretizing the GED can solve are analogous to those that can be solved in the continuous domain. The main distinction is in the ultimate modeling of acquired knowledge, where discrete states stimulate qualitative model inference whereas continuous values allow quantitative model inference (Misra & Ray, 2017). When compared to other methods that use continuous

values, the learning process using discrete data is more efficient and effective because it requires a less amount of data (Hu *et al.*, 2018). Furthermore, data reduction and simplicity speed up the learning process, resulting in more compact and shorter outcomes. Discretization approaches in GED can be divided into two groups which are unsupervised and supervised discretization (Anguita-Ruiz *et al.*, 2020).

### 2.11.1 Unsupervised discretization

In unsupervised discretization, no class label information is used in the computation of the discrete states of the genes that need to be provided by the user. Only GED is used to calculate the discrete values. Anguita-Ruiz *et al.*, (2020) classified these techniques based on as either supervised or unsupervised. The first is 'discretization utilizing absolute values,' which may be used to any GED because it directly discretizes absolute gene expression values using several methodologies. The second method is called 'discretization utilizing expression variations between time points,' and it only works with time series expression data, computing variations between each pair of consecutive time points (Gallo *et al.*, 2015).

#### 2.11.1.1 Discretization based on metrics

The metrics-based approaches compute the cut points  $P$  for the gene  $g$   $I$  in  $A'$  so as to determine those corresponding discrete state using a measure. These approaches follow a basic concept given in Equation 2.34 where the goal is to discretize the matrix  $A'$  with discretization level of two values (0,1).

$$a_{ij} \begin{cases} 1 & \text{if } a_{ij}' \geq \delta \\ 0 & \text{otherwise} \end{cases} \quad (2.34)$$

As shown in equation 2.34, binary matrix is made up of two symbols: one for 'activation' and one for 'inhibition' (for example, 1 and 0 as in Equation above). The most straightforward method is to define as the average expression value of a given data scope.

### 2.11.1.2 Discretization based on ranking

This method works by sorting all the expression values on a list  $L$  in decreasing order. The first  $x\%$  percent of  $L$  values are assigned 1, while the remaining values are assigned value 0. This is a basic method whose top percent  $\% X$  is the name given to this method.

### 2.11.1.3 Discretization based on clustering

Each value  $a'_{ij}$  of the GED  $A'$  is treated as an element of a single-dimensional space in this manner. The  $S$  elements of  $\Omega$  that correspond to a certain 'data scope' are then subjected to a clustering procedure. A gene profile, a column profile, or a matrix profile are used to get value of groupings where the values in each group are allocated to the same discrete state. Groups are created by maximizing similarity between components in each cluster while limiting similarity between items in different clusters. Within-Cluster Sum of Squares (WCSS) is a standard quality metric for clusters, which is defined as follows for a given discretization scheme  $D$ :

$$WCSS(D) = \sum_{a'_{ij} \in [p_0, p_1]} |a'_{ij} - \mu_0|^2 + \sum_{r=1}^{k-1} \sum_{a'_{ij} \in (p_r, p_{r-1})} |a'_{ij} - \mu_r|^2 \quad (2.35)$$

Where  $\mu_r$  is the mean of the  $a'_{ij} \in (p_r, p_{r-1})$

WCSS is calculated by adding the squared Euclidean distance between items within a cluster to the cluster's mean, with lower values indicating greater cluster element similarity.

### 2.11.2 Supervised discretization of gene expression data (GED)

Supervised approaches are rarely used in discretization of RNASeq data since unsupervised approaches have been devised for dealing with GED discretization. However, some approaches employ supervised algorithms and, in general, take prior biological knowledge into account when completing the discretization.

### **Supervised discretization approach.**

Supervised” discretization methods take the class into account when setting discretization boundaries and works as follows:

Given a GED matrix  $A'$  with  $N$  genes and  $M$  conditions, a set of classes  $\Gamma$ , and a matrix  $C$  (with the same dimensionality as  $A'$ ), set of classes are created.  $A'$  and  $C$  are the inputs to a supervised discretization technique, where  $C$  converts each  $a'_{ij}$  of  $A'$  into a target class label  $c \in \Gamma$ . A supervised strategy will aim to find a discretized matrix  $A$  that best matches the continuous expression values of  $A'$  with the target class label information of  $C$ . The number of classes will determine the level of discretization in this way. A k-means clustering algorithm was employed by Liu *et al.*, (2021) to discretize continuous data, and the apriori algorithm was used to do association analysis. This discretization method enables obtaining of strong association rule that has been validated, as well as the strong association rule's association degree and value interval. Results also reveal that the SAR's association connection and association degree are directly related to the value interval of characteristics, rather than being fully unaffected (Liu *et al.*, 2021).

Elhilbawi *et al.*, (2021) studied the importance of discretization as a preprocessing step that aids classification performance when compared to continuous characteristics. For the challenge of forecasting Intensive Care Unit (ICU) mortality, the authors investigated the effectiveness of numerous parametric and non-parametric discretization methods in combination with several machine learning classifiers. The importance of discretizing the input qualities in this challenge is demonstrated by their findings. The classification accuracy and F1 score for discretized data were 89.19 percent and 0.38, respectively, whereas the classification accuracy and F1 score for continuous attributes were 86.19 percent and 0.08. These findings show that discretizing continuous attributes before using machine learning models can improve performance dramatically (Elhilbawi *et al.*, 2021).

Sawangarreerak & Thanathamthee (2021) used discretization to partition the data range. The best method for discretizing was to use equal-width bins with five bins. To uncover

linked patterns, these ranges were combined with association rules and FP-Growth. They discovered trends that pointed to symptoms of bogus financial items, which financial statement users should pay attention to (Sawangarreerak & Thanathamthee, 2021). Miswan *et al.*, (2021) offered a framework that includes data management, such as data discretization, binary translation, and data balancing before moving to rule mining using ARM. They used supervised rule learning settings, such as readmission kinds and fundamental demographic variables, to extend ARM's medical application in hospital readmission of heart failure disorder. In terms of theoretical time complexity of the total processing framework, the processing stage spends the greatest time in relation to the magnitude of a dataset. When employing the Apriori technique from the 'arules' package, the rule mining extraction process is easier, and the difficulty is dependent on the number of input variables in the dataset (Miswan *et al.*, 2021).

Sari *et al.*, 2021 conducted a comparison experiment on three discretization methods: equal-width, equal-frequency, and K-means. The authors showed that the maximum level of accuracy was attained when the K-means algorithm was used to classify three continuous variables using the Bayesian networks model (Sari *et al.*, 2021). Bat-KMeans technique was introduced by Mohamed & Samsudin (2021) as a feature selection approach for finding the best feature from an optimized discrete dataset in order to reduce data dimension. Their experiment compares the classification effectiveness of discretization and feature selection to continuous datasets without feature selection, discrete datasets without feature selection, and continuous datasets without discretization and feature selection using the k-Nearest Neighbor approach. Bat is also shown to be useful as a feature selection and discretization approach. The experiments employed many benchmark datasets from the UCI machine learning repository. Findings show that Bat-KMeans optimized discretization and Bat-optimized feature selection increase classification accuracy (Mohamed & Samsudin, 2021).

Dhalmahapatra *et al.*, (2020) used a fuzzy discretization strategy, t-SNE technique, and fuzzy c-means clustering to improve the standard multiple correspondence analysis (MCA). This fuzzy discretization approach converts them to categorical variables o make



continuous variables analyzable with MCA, An R2-profile is used to get the most concealed dimensions that represent the most category information. t-SNE technique is then used to show the significant categorical correlations by representing the high-dimensional categorical information in a 2D map. The categories are then divided into various clusters using fuzzy c-means clustering (FCM) based on their membership degree. An ideal number of clusters is calculated using cluster validity indices. FCM findings were compared to those produced using the K-means (KM) algorithm and unsupervised fuzzy c-means clustering (UPFCM). On the basis of solution quality, FCM surpasses KM and UPFCM (Dhalmahapatra *et al.*,2020).

Fikri *et al.*, (2020) three simulations were run in their investigation, each with a distinct fuzzy discretization output. There are three types of fuzzy discretization outputs: 1) no fuzzy discretization, 2) fully fuzzy discretization, and 3) partial fuzzy discretization. The classification accuracy of the Random Forest classifier was observed, recorded, and assessed for all simulation versions. Addition of fuzzy discretization to random forest classification algorithms enhanced classification accuracy. The use of fuzzy discrete intervals on all attribute values, on the other hand, can result in a loss of classification accuracy for the random forest classification algorithm due to "over-discretization." Applying fuzzy discretization to only continuous variables and keeping the other attributes with their original discrete values can increase random forest classification accuracy when compared to translating all attributes into discrete fuzzy interval values. The authors suggested only using fuzzy on discretized attributes that are recognized and picked from continuous attribute values to boost classification performance (Fikri *et al.*, 2020).

Hranisavljevic *et al.* (2020) developed DENTA (Deep Network Timed Automaton), a unique machine learning discretization strategy that tackles the issues associated with discretization algorithm by constructing a deterministic timed automaton from the original mixed data. First, it uses a deep network of stacked restricted Boltzmann machines to extract new features from continuous input in a hierarchical manner (RBMs). They showed that high-level RBM abstractions may be used to automatically detect significant discrete events in continuous system behavior. Finally, as a discrete representation of

overall system behavior, a timed automaton is generated, allowing for a joint timing examination of the entire system. The model is verified on a synthetic and real-world dataset using anomaly detection, with the results proving the approach's clear advantages for a specific class of systems (Hranisavljevic *et al.* 2020).

## **2.12 Machine learning**

Machine Learning (ML) is the automated computational method with capability of discovering hidden as well as non-obvious patterns in a dataset using statistical methods implemented (Xu, 2019). Machine learning (ML)-based methods have been used in addressing the challenges associated with high dimensional big data from life sciences. ML have facilitated recognition, classification, and prediction of big biological data patterns (Li & Chen, 2014). Machine learning methods are categorized based on the way they learn from the data. These approaches are categorized as either supervised or unsupervised. Supervised learning works by classifying objects in a pool using known features or attributes. Supervised algorithms first learn the pattern from a subset of training data and then use the acquired knowledge to classify the remaining test data. In unsupervised learning, patterns are defined using a subset of unknown i.e., the algorithms start by defining the objects in a pool of data with unknown features or attributes and then using the acquired knowledge, they perform classification for the remaining data (Mahmud *et al.*, 2021).

Most machine learning algorithms for classification were developed with an assumption that there is an equal number of samples. Any imbalance in the classes leads to difficulties in predictive modeling (Bader-El-Den *et al.*, 2018). As a result, poor prediction accuracy is reported especially for the minority class (Elsakaan & Amroun, 2021). A problem arises since the minority class is usually more important, making the situation more susceptible to classification of majority by minority class. This situation makes machine learning models to become “lazy” in learning on how to discriminate among classes. In the end, the ML models favor the majority class, and this leads to synthetic high accuracy (Douzas *et al.*, 2018; Sarkar *et al.*, 2020).

Approaches towards addressing class imbalance are classified into three categories: data level, algorithm, and cost-sensitive approaches. Data driven techniques are more widely accepted because they do not rely on any algorithm and are flexible in integrating other techniques. Data driven approaches include under-sampling and oversampling (Zhu *et al.*, 2017) and are more capable of generating a balanced dataset. Oversampling works by increasing the minority class instances randomly or by replicating the same data through simulation techniques to improve the imbalance ratio. The main strength of this approach is that there is no loss of important information from the dataset. This is because the original dataset is usually retained although there is an addition of information to the data for the purpose of balancing. The major limitation of this approach is an increase in execution time because of the increased instances. In the under-sampling approach, the instances from the majority class are usually removed randomly using a defined criterion before classification. This approach is very simple but, in most cases, it leads to loss of some important information from the data (Kaur & Gosain, 2018).

### **2.12.1 Supervised learning**

Supervised learning is defined as a learning process where the system is guided (either automatically or by human interaction) and receives feedback about the correctness of its performance. In this type of paradigm, the performance measure  $P$  allows the system to improve its learning process continuously. The classification problem of the function is exemplified by supervised learning systems such as spam classifiers, face recognizers on images, and medical diagnostic systems for patients, where the training data take the form of a collection of pairs  $(x, y)$  and the goal is to produce a prediction and  $y^*$  in response to a query  $x^*$ . The  $x$  elements can be simple vectors or more complicated things such as texts, photos, DNA sequences, or graphics. Machine learning techniques have been applied to make the integration of different proteomic and genomic information easier (Gunaratne *et al.*, 2021).

### 2.12.1.1 Support vector machines

Support vector machines (SVM) is one of the most popular supervised learning algorithms that is used in literature. This algorithm is a non-probabilistic binary linear classifier that works by assigning unseen samples of data to one of two possible classes using a linear decision boundary. SVM learns a mapping based on training samples which maximizes the distance between two classes. Ideally, these classes become linearly separable. Nevertheless, SVMs can perform non-linear classification by mapping (Huysmans, 2021). SVM and kernel methods have been applied in prediction of protein-protein interactions. Experimentally determined interactions are analyzed to find patterns that distinguish the sequences of interacting protein pairs from non-interacting pairs (Zahiri *et al.*, 2013). These predictions are based on protein information such as physicochemical properties of the protein, structural information, evolutionary information, domain information etc. For example, protein domains can be identified within sequences and matching pairs of domains found to be enriched among known interacting proteins pairs can be used in the prediction of new interactions (Zahiri *et al.*, 2013).

Mazumder & Veilumuthu, (2019) proposed a feature selection approach using Joe's normalized mutual information on microarray cancer datasets. They compared five classifiers and reported an average increase in the prediction accuracy of 5.1% when feature selection was done before classification. Ray *et al.*, (2016) used a Microarray Leukemia dataset and proposed a three-step approach that involved: data preprocessing and normalization, feature selection using mutual information method and classification using SVM and regression analysis. They reported an improved computation time and efficiency of both classifiers although SVM performed better than logistic regression in terms of accuracy. Lokeswari & Jacob (2017) performed classification before and after application of feature selection on a Microarray pediatric tumor dataset. They reported that application of feature selection before classification improved the accuracy of the results on both SVM and logistic regression. However, SVM achieved accuracy of 75%, compared to 63% accuracy by logistic regression. Hasanin *et al.*, (2019) used MapReduce for feature selection on Protein Structure Prediction dataset followed by SVM, logistic

regression, and Naïve Bayes with and without feature selection. Analysis of the performance and running time showed that SVM outperformed the other classifiers (Hasanin *et al.*, 2019).

Alghunaim & Al-Baity (2019) used SVM, decision tree, and random forest algorithms to analyze gene expression and DNA methylation datasets to predict breast cancer. An experiment done using WEKA showed differences in terms of accuracy for both datasets where SVM achieved 98.03%, decision tree 95.09 and random forest 96.07 for gene expression data. Accuracies on methylation datasets were 98.03 for SVM, 88.23% for decision tree and 95.09% for random forest classifiers. This shows that SVM achieved the highest accuracy in both datasets. Turgut *et al.*, (2018) used RFE and RLR (Randomized Logistic Regression) on cancer dataset described in Matamala *et al.*, (2016) for feature selection. They thereafter applied SVM, KNN, MLP, DT, RF, LR, Ada and GBM classification models on the selected features. SVM gave an accuracy of 99.23% using both RFE and RLR as compared to 98.49% before feature selection. Morovvat & Osareh, (2016) used Symmetric Uncertainty (SU) filter methods and then applied CFS, FCBF, GSNR, ReliefF and MRMR feature selection to further reduce the number of attributes. Thereafter SVM, J48 decision tree and Naïve Bayes were used for classification. SVM gave the best results as compared to the other classifiers.

#### **2.12.1.2 Naïve Bayes Algorithm**

Simple probabilistic classifiers based on the Bayes theorem and strong independence assumptions between features are known as NaïveBayes classifiers. NaïveBayes is a simple method for creating classifiers, which are models that assign class labels to issue situations represented as vectors of feature values, using a limited number of class labels. It refers to a group of strategies for training such classifiers that are all based on the same premise. Given the class variable, all Naïve Naïve Bayes classifiers assume that the value of one feature is independent of the value of any other feature (Granik & Mesyura, 2017). Nagarajan *et al.*, 2019 used three classifiers which were SVM Naïve Bayes and LDA to predict the availability salivary matrix metalloproteinase- 8, serum biomarkers. Predictive

model using Naïve Bayes Classifier was able to identify progressors with sensitivity of ~89% (Nagarajan *et al.*, 2019).

In classifying ageing-related genes based on real data, researchers utilized NaïveBayes classifiers as base models, with uncertain features indicating protein-protein interactions. The authors reported that their model experimental results which was an ensemble of NaïveBayes Classifiers provided a better prediction performance as compared to a single NaïveBayes classifiers and also conventional ensembles (de Holanda *et al.*, 2021).

### **2.12.1.3 Multilayer perceptron**

Multilayer Perceptron (MLP) belongs to a class of feedforward artificial neural networks, which find complex patterns that a human programmer cannot extract by performing machine recognition. In their study, Eluri (2021) used Keras modeling on MLP to discover important features from a gene expression dataset. After initial training with top returning features from training classifiers, MLP retrieves features from the test datasets. Finally, MLP is fine-tuned to extract optimal features from gene expression datasets using top returned features, specifically GENT2. In terms of accuracy, f-measure, precision, and recall, results suggest that MLP extracts features better than other approaches.

Seo & Cho (2020) suggested a feature selection approach called ‘boosted regression-based feature selection’ for the multilayer perceptron (BREG-MLP). BREG-MLP repeats the boosted feature selection procedure to obtain the smallest feature subset while maintaining outstanding classification performance.. The authors tested the proposed BREG-MLP on certain human cancer-related gene expression data sets in order to extract significant features, and the findings were that it performed better than single regression-based feature selection methods. Rawat *et al.*, (2018) proposed an advanced machine intelligence technology for predicting persistent respiratory disorders in children. Nine of the forty-eight extrapolative characteristics of the tenacious respiratory illness were found using discriminate partial least square regression. The most effective estimate accuracy is

urged by multilayer perceptron setups. These findings indicate that the approach is capable of accurately predicting asthma outcomes at 99.77%.

Desai & Shah (2021) examined artificial neural networks, MLP and CNN, which were used to detect breast malignancies for early breast cancer detection to see whether method was superior for diagnosing breast cell malignancies based on accuracy. Deep comparisons of each network's functioning and design were made, and then analysis was done based on the network's accuracy in diagnosing and classifying breast cancer to determine which network outperforms the other. In the diagnosis and detection of breast cancer, CNN was found to be more accurate than MLP. There is still a need to do a complete analysis and research using both methods on the same data set under the same conditions to determine the design that provides superior accuracy (Desai & Shah, 2021).

#### **2.12.1.4 Deep Learning**

Deep learning is one of the most effective machine learning techniques among the available approaches (Goodfellow *et al.*, 2016). Its main areas of use are image recognition and speech recognition where it has set records for other machine learning methods. Since deep learning algorithms are exceptionally effective at identifying complex structures in high-dimensional data, they have great potential in a wide range of other scientific fields, particularly precision medicine and genomics data processing. Deep learning techniques, however, are still extremely new to the bioinformatics field, and have found a major use in classification of genes based on their expression profiles (Daoud & Mayo, 2019). Unlike images or text data, gene expression data does not have a structure that can be utilized in the construction of a neural network. Multilayer perceptron is an architecture utilized for gene expression data prediction (Basavegowda & Dagnev, 2020; Guo *et al.*, 2017). Fakoor *et al.* (2013) employed a stacked autoencoder and principle component analysis (PCA) to minimize the dimension of the data prior to building a neural network for cancer prediction. Dincer *et al.* (2018) first used PCA, followed by a variational autoencoder for dimensionality reduction and a LASSO to predict the response to a leukemia therapy. Hanczar *et al.*, (2018) pretrained each layer of a multi-layer

perceptron using a denoising autoencoder and a sizable unlabelled dataset to predict malignancies. In particular for deep learning models, interpretation of machine learning algorithms is still a developing topic of study (Samek *et al.*, 2020). Prediction and model interpretation are two different interpretations that can be distinguished (Chakraborty *et al.*, 2017; Guidotti *et al.*, 2018). Model interpretation describes reasoning behind the model while forecasting various outputs on the entire population. Prediction interpretation involves describing the prediction of a given input. However, for medical uses, both are crucial.

There hasn't been much research done on the interpretation of neural networks created from gene expression. Majority of previous studies concentrate on determining which genes had an impact on the prediction but do not look into how the representation of the learned gene expression in the hidden layer is represented. For instance, Danaee & Ghaeini *et al.*, (2017) used stacked denoising autoencoders to identify important genes for the diagnosis of breast cancer. The Kyoto Encyclopedia of Genes and Genomes (KEGG) and Gene Ontology are used to analyze the relevant genes, which are those that have a highly transmitted influence on the network's reduced dimension (GO). Finding potentially intriguing genes linked to the condition of interest is the goal of all these investigations. They do not, however, describe the function of the network, what a neuron represents, or how the patient is portrayed in the hidden layers.

Deep learning architectures have been applied in several studies in the literature to analyze biological data. Convolutional neural networks (CNN) were used in analysis of breast images to find mitosis and in segmentation of the brain tumor as well as MRI in neuroendocrine carcinoma detection (Le *et al.*, 2019; Havaei *et al.*; 2016 Noor *et al.*, 2019; Bi *et al.*, 2020). Wekesa *et al.*, (2020) proposed a GPLPI which was a graph representation method to facilitate prediction of plant long non-coding RNA-protein interaction (LPI) by use of sequence and also structural information. This model employed long short-term memory (LSTM) coupled with graph attention. This model was trained and tested using two datasets which were *Arabidopsis thaliana* and *Zea mays* datasets. The authors reported accuracies of 85.76% and 91.97% respectively *et al.*



Putin *et al.*, (2016) proposed a deep learning-based framework for chronological age prediction. They used an Ensemble-based combination of deep neural networks (DNNs) that were trained using blood biomarkers. A variation of the permutation feature importance was employed in this study to evaluate importance of each blood markers to ensemble accuracy. The best performance reported in this study was by a DNN with a mean absolute error (MAE) of 6.07 years in prediction of chronological age. The results of the ensemble-based learning provided an MAE of 5.55 years.

Bobrov *et al.* (2018) proposed using eye corner pictures to predict BA using a DNN-based model PhotoAgeClock. Their method yielded an MAE of 2.3 years and a 95% correlation with CA; however, BA was not considered. Mamoshina *et al.*, (2018) employed a multilayer DNN model to reveal population-specific aging patterns in Canadians, Koreans, and Eastern Europeans. Rahman & Adjeroh (2019) used a deep convolutional long short-term memory (ConvLSTM) model to estimate BA on a week's worth of physical activity data recorded per minute. Another study that used deep learning on biological data was done by Mamoshina *et al.*, (2018) who employed a multilayer DNN model in revealing aging patterns of the population. Rahman & Adjeroh (2019) also applied a deep convolutional long short-term memory (ConvLSTM) model which helped in estimation of biological age using physical activity data that was recorded every minute, to estimate BA on a week's worth of physical activity data recorded per minute (Pyrkov *et al.*, 2019). Recent advancements and opportunities in employing artificial intelligence (AI) for aging and longevity research were reviewed in (Miotto *et al.*, 2019). They examined works on DL, transfer, and reinforcement learning. Even though this was a thorough study on aging and lifespan that described machine learning (ML) techniques utilized in many aging studies, the publication did not address the issue of quantifying BA. Ashiqur *et al.*, (2021) examined deep learning algorithms based on several forms of bioinformatics data. The DL models DNN, CNN, ConvLSTM, and CNN+LSTM were trained to exploit the physiological/activity changes' reliance on age. The DL methods were trained to reduce the MSE between the chronological age and the estimated biological age in all circumstances (Ashiqur *et al.*, 2021).

Some studies have been done on deep learning in mining motifs. The impact of various parameters in deep learning, such as the number of layers, on motif mining was studied (Zhang *et al.*, 2016). Other researchers made more attempts at deep learning frameworks, adding an LSTM layer to DeepBind and obtain a novel model for motif mining that could combine CNN and RNN (Quang & Xie, (2016) .Recurrent Neural Networks(RNN) and CNN models have been combined to gain the advantage of both models in classification. Addition of RNN layer facilitates the ability to capture dependencies between sequence features by learning the features recovered by the CNN layer, thus improving prediction accuracy (He *et al.*, 2021).

A number of studies have attempted to understand the hidden neurons, and practically all of them are based on an analysis of the values or connection weight distribution of the learnt neural network (Teixeira *et al.*, 2017). Way *et al.* (2018) examined the connections between decoders and their variational autoencoder and linked each neuron to the group of genes with the highest absolute values of weight. They used an enrichment analysis to find overrepresented pathways and GO biological process terms for each neuron based on these gene sets. Way & Greene's (2018) constructed denoising autoencoders and stacked denoising autoencoders to extract significant genes from a dataset of cancer gene expression. . The total number of connections that genes have outwardly determines their significance. The importance of genes is defined as the sum of their outgoing connections. A functional annotation analysis is conducted on a subset of the most significant genes before being further examined. In order to determine their significance for the prediction of the metastatic tumor, Sharifi *et al.* (2019) examined the distribution of each neuron's output weights.

Deep learning's lack of interpretability in medical applications is one of the primary worries. Neural networks can be thought of as "black boxes," where a patient's gene expression profile is fed into one layer and a prediction is drawn from another without any explanation of the decision-making process. The demand to make deep neural networks more interpretable is growing, notably in the medical area. It's important to make sure a neural network does not focus on a data artifact and instead its predictions is based on

trustworthy representations. Physicians cannot trust the neural network's judgment without the interpretability requirement being satisfied, and patients' lives could be in danger. Knowing which neurons, genes, and other biological processes are involved in prediction and decision-making is essential. In addition, a neural network with strong prediction abilities might have discovered patterns in gene expression that might inspire fresh biological concepts. Understanding the biological significance of the network's hidden layers is essential to examining these patterns.

### **2.12.2 Unsupervised**

Unsupervised learning usually entails analyzing unlabeled data while making assumptions about the data's structural features (for example, algebraic, combinatorial or probabilistic). The data are assumed to be in a low-dimensional variety, and the goal is to explicitly identify that variety from the data. Principal component analysis, multiple learning, factor analyses, random projections, and automatic coders are examples of dimensional reduction techniques. A variety of grouping processes have been devised, all of which are based on certain assumptions about what constitutes "clustering." Because the goal is to use the unusually large data sets that are available if supervised labels are not used, computational complexity is crucial in both grouping and dimension reduction (Karim *et al.*, 2021).

#### **2.12.2.1 Association Rule Mining**

Association rule mining (ARM) is a market basket analysis algorithm described by Agrawal *et al.*, (1993). It is a data mining approach that has been widely used to discover high frequency co-occurrence of items in databases. In ARM, datasets are presented in a transaction format whereby a transaction  $t \in \mathbf{D}$  contains itemset  $\mathbf{X} \subseteq \mathbf{I}$  if  $x \subseteq \mathbf{T}$ . ARM has been successfully applied in many areas like market basket analysis, health care and in recommender systems (Viktoratos *et al.*, 2018), classification of cancer gene expression data (Alagukumar & Lawrance, 2016) and identification of malignant mesothelioma risk factors (Alam, 2019). Quality measures are used in selecting the best sets of the generated

frequent patterns. Probabilistic measures have been applied to evaluate the generality and the reliability of association rules. Support measures the generality of the rules while confidence and lift are used to show reliability of the rules. Support and confidence are some of the widely used measures in evaluating the quality of the rules (Telikani *et al.*, 2020).

A unsupervised method of association rule mining (ARM) is used to uncover patterns in massive datasets. Agrawal *et al.* developed this algorithm in 1993 with an initial application's goal being to detect patterns of items purchased together in transaction databases. Items are treated as strings in that application, and they might be present or absent in a single transaction (Altaf *et al.*, 2017).

A nondeterministic polynomial is used to find Association Rules (ARs) in a transaction database which is a NP-Hard problem. If the dataset has  $n$  items, the number of itemsets is  $2^n - 2$  is the maximum number of ARs that can be recovered from each itemset, where  $k$  is the itemset length. Apriori-based algorithms have a time complexity of  $(2^n) + O(2^k)$ . As a result, discovery of ARs has an  $O(k \times 2^n)$  time complexity (Almasi & Abadeh, 2015). This shows that as the number of items increases, running time increases exponentially (Beiranvand *et al.*, 2014).

Traditional ARM methods necessitate a significant amount of processing time. Furthermore, they rely on data preparation prior to executing the algorithm, which results in information loss. In fact, traditional ARM approaches have two drawbacks: a strong boundary between intervals in numeric characteristics and differentiation of the degree of membership for the interval in fuzzy sets. The purpose of association rule mining algorithm's is to find a set of rules that are above user-specified support and confidence thresholds. The first stage is to identify all 'frequent' itemsets those with a support level over the threshold. The frequent itemsets are then used to construct association rules and any rules that fall below the confidence threshold are removed. Enumerating the most common itemsets is the most complex step since the desired itemsets have to be found among the total  $2^{|U|} - 1$  itemsets which can actually be generated (Telikani *et al.*, 2020).

### 2.12.2.1.1 Apriori

The Apriori algorithm was developed to work with transactional databases. Apriori employs a "bottom up" strategy, in which frequent subsets are expanded one item at a time (a process known as candidate generation), and groups of candidates are evaluated against the data. When no more successful extensions are detected, the algorithm ends. Apriori uses a Hash tree structure and a breadth-first search to effectively count candidate item sets. From item sets of length  $k - 1$ , it generates candidate item sets of length  $k$ . The candidates with an infrequent sub pattern are then pruned. The candidate set comprises all frequent  $k$  length item sets, according to the downward closure. The system then analyzes the transaction database for common item sets among the candidates (Agrawal & Srikant, 1994).

An itemset lattice is a set of all itemsets that may be created from a given dataset, in which itemsets are linked by subset/superset relationships. To locate all the frequent itemsets in the lattice, the Apriori method employs a breadth-first search strategy. First, all size 1 itemsets are counted. Any superset of an infrequent itemset will also be infrequent, therefore itemsets with support below the threshold are deleted. The leftover size 1 itemsets are used to generate candidate size 2 itemsets, after which infrequent size 2 itemsets are destroyed. This process is repeated until no frequent itemsets of size  $n$  exist, in which case candidates are generated from frequent itemsets of size  $n - 1$  and infrequent itemsets are discarded

After the itemset mining is completed, the rules are generated. As an example, depending on the itemset  $B, E$ , we might want to generate the rules  $B \Rightarrow E$  or  $E \Rightarrow B$  that is based on based on the itemset  $\{B, E\}$ . According to these criteria, if one of the two goods appears in a transaction, the other is quite likely to appear as well. The rule  $B \Rightarrow E$  has a 100% confidence level because every transaction that contains  $B$  also contains  $E$ , but the rule  $E \Rightarrow B$  has a 75% confidence level because  $E$  appears in four transactions but  $B$  in only three. While Apriori's method is straightforward it simply requires the union of sets of

items and scans over the set of instances to validate support, it is inefficient it prunes itemsets with are considered infrequent in subsets (Ryan, 2016).

#### **2.12.2.1.2 Eclat**

Equivalence Class Clustering and bottom-up Lattice Traversal are abbreviated as Eclat (Zaki,2000) This approach decreases these costs by using a vertical transaction ID set database format, equivalence class clustering, and bottom-up lattice traversal. Eclat converts horizontal databases to vertical databases, i.e., from  $\langle TID_i, i_1, i_2, \dots, i_k \rangle$  to tidset format  $\langle i_k, TID_1, TID_2, \dots, TID_k \rangle$ . Each transaction  $T_i$  in a horizontal database has a unique identifier  $TID_i$  and an itemset in the form of  $\langle TID_i, i_1, i_2, \dots, i_k \rangle$ . The TIDset of an item or itemset  $X$  is the set of all transaction identifiers containing  $X$ , and is denoted as  $\text{tidset}(X) = \{T_i.TID | T_i \in D, X \subseteq T_i\}$ . Support of an item or itemset  $X$  is the number of elements in  $\text{tidset}(X)$ . For example,  $\sigma(X) = |\text{tidset}(X)|$ . An itemset  $X$  is said to be frequent if  $\sigma(X) \geq \text{min\_sup}$ , where  $\text{min\_sup}$  is specified by the user as a minimum support threshold (Dong & Liu *et al.*, 2015).

The intersection of tidsets of a candidate  $k$ -two item set's  $(k - 1)$  -subsets yields its support. The vertical database is smaller than the horizontal database and contains all necessary information, reducing memory needs and database scanning. In addition, as the length of itemsets grows, their tidset decreases, lowering the cost of intersection operations. When calculating frequent 2-itemsets, the vertical format is more expensive than the horizontal format. To update the counts of candidate 2-itemsets, a triangular matrix is employed (Zaki, 2000)

#### **2.12.2.1.3 Frequent Pattern Growth (FP-Growth)**

Han *et al.*, (2000) proposed a strategy called FP-growth that completely avoids candidate generation. This method generates a frequent pattern tree (FP-tree), which keeps track of how often each item pattern appears. Because the nodes in the branches of a tree have an order, these patterns are sorted sets of objects rather than the unordered sets of things that

make up the underlying dataset. A tree can be browsed and managed to enumerate frequent itemsets without ever generating infrequent itemsets that must be eliminated. (Wicaksono *et al.*, 2020).

Furthermore, unlike the original dataset, a tree may represent frequency data in a more compact fashion. Each branch of the tree shows a common pattern. Beginning with the root node and ending with the node in question, each node in the tree corresponds to a single item and records the frequency of the pattern generated by it and its ancestor nodes. Because an item may appear in a variety of patterns, it may have numerous nodes, with the item's overall frequency equal to the sum of the frequencies recorded in all of the nodes. A linked list connects nodes that refer to the same item, making traversal of all nodes that refer to the same item more efficient (Shabtay *et al.*, 2021). The most often occurring item becomes a child of the root node when a new instance is added to the tree, the second most frequently occurring item becomes a child of the first item, and so on. When patterns in the tree with the same prefix overlap, the frequency recorded in the current nodes rises until the patterns diverge, at which point the new pattern's branch splits from the existing branch. This is what allows the tree to be a smaller representation of the frequencies than the original dataset, as well as removing infrequent items before generating the tree (according to the same principle used by Apriori); overlapping patterns reduce duplication and save space. Enumerating the frequent itemsets necessitates a recursive traversal of the tree.

The most common items are chosen in reverse order of frequency (i.e., the reverse of the order used when building the tree). A conditional FPtree is produced for each item, which is effectively an FPtree made up of only those instances that include the target item. This is accomplished by traversing the main tree and collecting patterns that contain a target item, then constructing an FPtree from the extracted patterns. Items in the conditional FPtree that are common are those that appear frequently with the target item, resulting in frequent 2-itemsets. (Wicaksono *et al.*, 2020).

The conditional FP-tree method is done recursively for each of these frequent itemsets in order to find frequent itemsets involving those itemsets until no more frequent itemsets arise. The operation continues, ignoring any item that was already processed, by selecting the next item from the initial reverse-order list. All the common itemsets are enumerated in this fashion, utilizing the FP-tree structure, without the need for candidate generation and only two scans of the source dataset, while the dataset is represented in a more compact format (Ranjan & Sharma, 2019).

#### **2.12.2.1.4 Association rule mining on biological data**

As the sets of gene expression data became increasingly large, data mining techniques have become crucial to analyze expression data. Many grouping techniques have been explored to group genes based on similar expression profiles (Alagukumar & Lawrance, 2016). Looking for association rules in the data is a common data mining technique that, unlike clustering, is used to uncover and explain links between distinct items in a large data collection. The format of an association rule is  $LHS \Rightarrow RHS$ , where LHS and RHS are sets of items, and the RHS set is likely to occur whenever the LHS set does. The rules of association are known as "market basket analysis" in the retail industry (Sagin & Ayvaz, 2018). An association rule represents a group of things that are likely to be purchased together in a market basket analysis; for example, the rule cereal, milk, juice would state that anytime a client buys cereal, he or she is likely to also buy milk and juice in the same transaction. In biological datasets, elements in an association rule can represent biological features that are strongly expressed or repressed in response to either exposure to certain treatments of when comparing a disease versus healthy condition. Public gene expression data large enough to search for association rules and obtain meaningful results are now available (Deelen *et al.*, 2019).

Nagata *et al.*, (2014) analyzed toxic genomic and toxicological data using classification association rules. They then reported that the statistical t test had been applied often in the analysis of microarray data. They then selected only those genes that were up-regulated (change of times > 2 and  $p < 0.05$ ) or down-regulated (change of change < 0.5 and  $p < 0.05$ )



in the groups with increased or non-decreased groups, respectively (Nagata *et al.*, 2014). Methods for analyzing gene connection using frequent patterns were discussed in earlier study by Alves *et al.*, (2010). Mining of common patterns has been effectively used to identify patterns of knowledge in a variety of data, including commercial and scientific data, and is emerging as a promising technique in the study of microarray gene expression. These methods, on the other hand, are poorly scaled and often not practicable with dense data sets such as telecommunications, microarrays, and so on, where there are numerous frequent and extended patterns. This issue arises because of the preceding algorithm's high processing expenses. Methods based on trees, such as FP growth, may encounter some difficulties when dealing with dense or high-dimensional data sets. Earlier study by Zakaria., *et al.*, (2014) proposed an enumeration of columns-based algorithm that uses high trust association rules for genes expressed up and down. They then explained that the generation of all sets of frequent elements in dense data sets requires a large memory (Zakaria *et al.*, 2014)

Nagata *et al.*, (2014) used the association method to classify toxicological data in earlier research. The changes in the folds and the p-values of the student's t test performed between a group of treated chemicals and their matching control group were used to discretize the genetic expressions and relative liver weights. The Apriori-TFP algorithm was used to generate class association rules. CBA is superior to LDA in terms of prediction performance and interpretability, according to the classification between CBA and A linear discriminant analysis (Nagata *et al.*, 2014)

Various cross validation approaches were explored by Refaeilzadeh *et al.*, (2009). Cross-validation is a statistical approach for calculating and comparing learning algorithms that divides data into two parts, one for learning or training a classifier model and the other for validating it. The associative classification algorithm comprises three stages: discretization, rule creation, and classifier construction (Refaeilzadeh *et al.*, 2009). Indhumathy *et al.* (2018) used a bi-clustering algorithm along with a new way of association rule mining to evaluate human HCV PPI data. The criteria based on Gene Ontology (GO) annotations were weighed using a mathematical model. As a result of this

strategy, past knowledge-based protein annotations might be used to make more educated predictions when applying association rules. Secondly, when a prediction criterion was applied, this strategy assisted to better comprehend the shared properties of proteins based on their GO annotations. The novel expected contacts of the HCV human protein were predicted using the newly found association rules, and some of the projected interactions were confirmed using a literature review. For validation purposes, further enrichment studies were conducted, including gene ontology analysis, route-based analysis, and disease association analysis. Human proteins that interact with HCV proteins in the projected network have similar biological functions, according to our analysis (Indhumathy *et al.*, 2018).

Shui & Cho (2016) suggested a rank-based weighted association rule-mining method for gene expression analysis. The authors claimed that by using this method, they were able to reduce the number of rules generated, as well as the duration and execution time. Gene ontologies were used to validate the rules developed using this method. Agapito *et al.*, (2015) conducted another study on association rule mining in gene expression data, and they were able to electronically infer annotations of association rules by assigning different weights to different forms of annotation.

### **2.13 Research gap**

Machine learning, feature extraction and feature selection are some of the strategies used in dimensionality reduction. They have been used in the analysis of different types of data. However, they have a major limitation in that they focus on evaluating each feature individually instead of putting into consideration the interactions or dependencies between features. These relationships are very important because they determine the functional/phenotypic aspects in living systems. Therefore, there is a need for an alternative model that will help not only select the informative features but also help predict the phenotype of unknown features based on their interactions with those with a known function.



## CHAPTER THREE

### METHODOLOGY

#### 3.1 Study design

Figure 3.1: below gives a summary of the study design in form of a workflow adopted in this research.

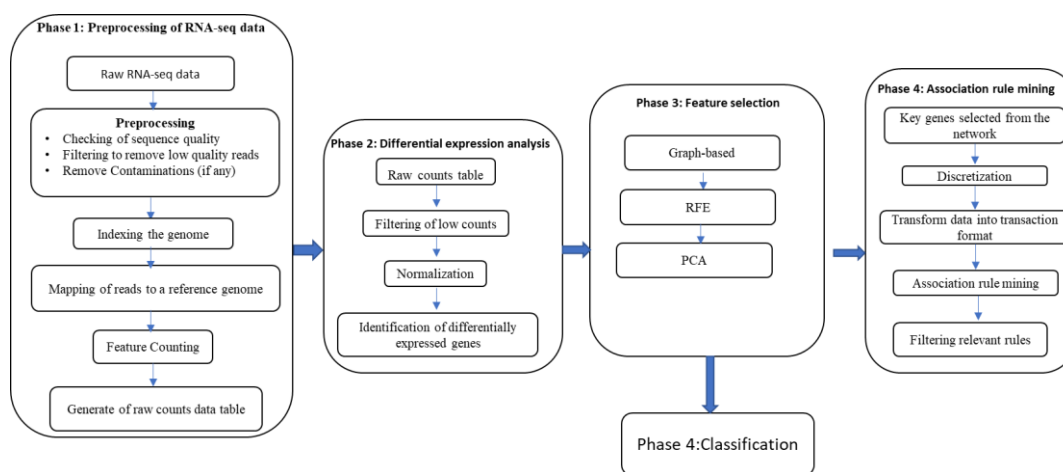


Figure 3.1: **Workflow of the study**

As indicated in figure 3.1, the first step was acquisition of raw RNASeq data from the SRA archive for analysis. After data acquisition, the quality of the reads was checked to ensure only reads above the cut-off quality score and free from contamination were retained for subsequent Three trimming algorithms were compared, and the output of the best performing trimming algorithm was used in the next step which mapping the reads to the reference genome. However, before this step, the reference genome was indexed to create a data structure of the same for purposes of efficiency in the mapping step. A genome is the complete set of genetic information in an organism which provides all the information the organism requires to function. The essence of mapping the raw RNAseq data to the reference genome is to get the identity of the expressed genes and also to facilitate counting the number of times each gene is expressed (expression level) in

response to a certain condition. The last step of preprocessing is to count the features/genes with the final output being a .CSV file of count data. The second phase as shown in figure 3.1 is filtering of the low counts or the features/genes have zero values. Normalization is a crucial step in RNASeq data analysis whereby raw data counts are adjusted taking into consideration characteristics such as differing sequence depth that would hinder direct comparison of expression values. Only the differentially expressed genes were retained for further analysis. Phase three was the feature selection phase. The proposed graph feature selection approach was used to filter and retain only related feature/genes from the differentially expressed genes. For comparison purposes, two other popular feature selection approaches which are RFE and PCA were also evaluated. Features selected using the three approaches were used to build classification models and the results compared. In the final phase, the count values of each differentially expressed feature/gene were retrieved from the normalized count table and discretized. Thereafter association rule mining was done and the strength of the rules generated using the three feature selection approaches evaluated. Details of each step is explained in the subsequent sections

### **3.2 Data type and data source**

RNA-Seq datasets used in this study are shown in Table 3.1. The first dataset is from the antennae of *Glossina morsitans morsitans* under the accession number PRJNA344035. This data set is from an ongoing project on Tsetse fly at KALRO and is comprised of 13 samples with the following classes: 4 control samples, 3 samples from flies exposed to a repellent, 3 samples exposed to an attractant 3 and 3 from flies fed using an artificial meal. A reference genome used at the mapping stage was downloaded from Vectorbase repository (<https://www.vectorbase.org>). The second dataset referred to as Small Cell Lung Cancer (SCLC) has 86 samples with two classes: 79 cancer cells and 7 normal cells. The third dataset denoted as Non-small Cell Lung Cancer (NSCLC) has a total of 218 samples whereby 199 samples are non-small cell lung cancer and 19 are normal cells. The gene annotation used was for *Homo sapiens* (GRCh38). The second and third datasets were downloaded from Gene Expression Omnibus (GEO) database

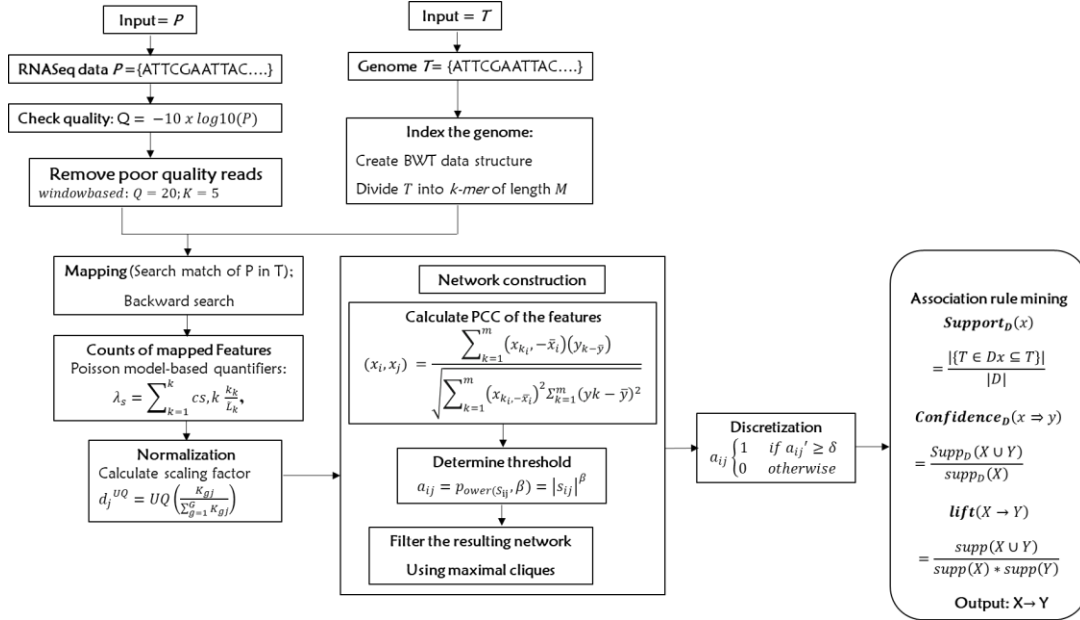
(<http://www.ncbi.nlm.nih.gov/geo/>). The Computing resources used in this study was 32 Cores (Each 8GB) = 256GB of RAM; 2 TB storage.

**Table 3.1: summary of datasets**

<b>Dataset Name</b>	<b>Instances</b>	<b>Attributes</b>	<b>Classes</b>	<b>Source</b>
SCLC (GSE60052)	86	28,089	2 (79 small cell lung cancer and 7 normal)	Jiang <i>et al.</i> , 2016
NSCLC (GSE81089)	218	28,089	2 (199 non-small cell lung cancer and 19 normal)	Djureinovic <i>et al.</i> , 2016
<i>Glossina morsitans morsitans</i> (PRJNA344035)	13	13,080	4 (4control attractant, repellent and fed)	3 3 3

### 3.3 Graph-based feature selection model algorithm

The proposed graph-based model for feature selection and phenotype prediction is depicted in figure 3.2. the model requires two inputs which is raw RNASeq data generated using next generation sequencing technologies (Input P) and a reference genome (Input T) (figure 3.2).



**Figure 3.2:** Graph-based model for feature selection and phenotype prediction.

The detailed explanation of every step in the graph-based feature selection model is described below.

### 3.3.1 Checking the quality

Data analysis began with preprocessing which is the most time demanding step in the process of RNAseq data analysis (Pérez-Rubio *et al.*, 2019). It involved quality control check, adapter trimming, contamination removal and quality filtering before transcript or gene quantification/counting (Figure 3.1). The Quality of the reads was checked by first assessing the base quality. The usual ASCII encoding is Phred+33. Phred is a base-calling program for DNA sequence traces which was developed by Dr. Phil and Brent Ewing (Ewing & Green, 1998). Phred reads DNA sequence chromatogram files and analyzes the peaks to call bases, assigning quality scores ("Phred scores") to each base call. The process of inferring the order of nucleotides (ATCG) in a template from the signals generated by the sequencing machine is referred to as base calling. After calling bases, Phred examines the peaks around each base call to assign a quality score to each base call. Quality scores range from 4 to about 60, with higher values corresponding to higher quality. Quality

scores are logarithmically linked to error probabilities, as shown in Table 3.2. Each quality score represents the probability that a corresponding nucleotide call is incorrect.

**Table 3.2: Error probabilities for assigning quality scores (Ewing & Green, 1998)**

Phred quality score	Probability of wrong base calling	Accuracy of base call
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1,000	99.9%
40	1 in 10,000	99.99%
50	1 in 100,000	99.999%

Phred's error probabilities have been demonstrated to be extremely accurate (Ewing & Green, 1998). If Phred gives a base a quality score of 40, the chances of this base being called wrongly are 1 in 10,000. This logarithmically based quality score is calculated as follows:  $Q = -10 \times \log_{10}(P)$ , where P is the probability that a base call is erroneous

We started by calculating the quality score using formula:  $Q = -10 \times \log_{10}(P)$ , where P is the probability that a base call is erroneous. First the p which is 0.01 according to table 3.1 was converted to exponential notation which is  $(1E^{-2})$

$$\text{Log}_{10} * (1E^{-2}) = -2 \tag{3.1}$$

Then  $-2 \times 10 = 20$  and this is a confirmation that the minimum quality score for the reads is 20.

### 3.3.2 Removing low quality reads

Trimming of low-quality ends was done based on the user specified parameters. With the minimum quality score set to be 20, features that did not meet the minimum threshold were filtered out. At this step, three techniques classified into three main categories were evaluated. These are:

1. Running sum algorithms
2. Window based algorithms



### 3. *K-mer* based algorithm

#### 3.3.2.1 Experiment 1: Running sum algorithms evaluation

Given a threshold value  $Q$ , the algorithm works in two steps. In the first step, the algorithm computes the first index  $l$  where the quality is greater than  $Q$ . In the second step, the program calculates  $S(l) = \text{quality}(l) - Q$  and the running sum:

$$S(i) = S(i - 1) + \text{quality}(i) - Q \quad (3.2)$$

When  $i$  is greater than  $l$ , the part of the sequence not trimmed is the region between the position  $l$  and the last position whose running sum is maximal. Everything before and after is trimmed. After that, if the good region length was lower than a threshold or if the mean quality in the good region was lower than a threshold, then the read was discarded.

#### 3.3.2.2 Experiment 2: Window based algorithms evaluation

A window spans the read from 5' to 3' and only bases at 3'-end are removed. Given a window's length and a quality threshold  $Q$  (the option *SLIDINGWINDOW* takes two parameters which are the window size and the minimum average quality, and it has no default values) the algorithm cuts the 3'-end when the average quality drops below  $Q$ . The minimum quality was set to be 20 for every window in both the forward and reverse reads.

#### 3.3.2.3 Experiment 3: *Kmer*-based algorithms evaluation

*K-mer* based algorithms work by comparing reads to the reference dataset  $k$ -mers enabling edit distance. Any read matching a reference *k-mer* is discarded. Once a reference *k-mer* is matched in a read, that *k-mer* and all the bases to the right will be trimmed, leaving only the bases to the left; this is the normal mode for adapter trimming. They also trim or remove those parts that match the reads instead of binning.

### 3.3.3 Experiment 4: Indexing and mapping to the reference genome

Genome indexing is the process of sub-setting the genome of length  $T$  into substrings of length  $s$ . Mapping is a very important step in analyzing any new generation sequence data. In all cases, the mapping process starts by building an index of the reference genome or the reads, which is then used to quickly retrieve the set of positions in the reference sequence where the reads are more likely to align. In this phase three indexing approaches were evaluated and they are categorised based on the data structure they use. These are algorithms based on hash tables; algorithms based on Burrows-Wheeler transform and algorithms based on Suffix array data structures.

### 3.3.4 Experiment 5: Counting of the features

The number of reads (counts) aligned onto each transcript were quantified in the respective BAM files using Salmon version 1.2.1 software (Patro *et al.*, 2017). Salmon algorithm uses the approach of *Kallisto* (Bray *et al.*, 2016) to compute the effective length of a transcript  $t_i$  defined as:

$$t_i = \ell_i - \mu_d^{\ell_i} \quad (3.3)$$

where  $\mu_d^{\ell_i}$  is the truncated fragment mean. The algorithm takes a maximum likelihood to look for the sizes of interest. With assumption that there is independence generation of all fragments, a known nucleotide fractions  $\eta$  is a binary matrix of transcript-fragment  $Z$  with  $z_{ij} = 1$ . The probability of observing of a set of sequenced fragments  $\mathcal{F}$  is defined as:

$$Pr\{\mathcal{F} | \eta, \mathbf{Z}, \mathcal{G}\} = \prod_{j=1}^N Pr\{f_j | \eta, \mathbf{Z}, \mathcal{G}\} = \prod_{j=1}^N \sum_{i=1}^M Pr\{t_i | \eta\} \cdot Pr\{f_j | \mathbf{t}_i, \mathbf{z}_{ij}\} = 1 \quad (3.4)$$

$|\mathcal{F}| = N$  is the sequenced fragments number,  $Pr\{t_i | \eta\}$  is the probability of choosing transcript  $t_i$  for generating a fragment given a fraction of the nucleotide  $\eta$ , and  $Pr\{t_i | \eta\} = \eta_i$  (Bray *et al.*, 2016). This provided data on relative read abundance by different treatments and replicates.

### 3.3.5 Experiment 6: Normalization

The resulting features were preprocessed to filter out any features with zero counts. This was achieved using the upper quartile normalization. A scaling factor of 75th percentile of every count was calculated after removing features with zero counts using the formulae:

$$d_j^{UQ} = \left( UQ \frac{k_{gj}}{\sum_{g=1}^G K_{gj}} \right) \quad (3.5)$$

Where  $UQ(X)$  is the upper quartile of sample  $X$  of  $j$ th sample of normalized counts and  $K_{gj} > 0$

#### 3.3.5.1 Experiment 7: Differential expression analysis

After normalization, differential expression analysis was done on the gene count matrix generated by Salmon (Patro *et al.*, 2017) using DESeq2 R-package version 1.28.0 (Love *et al.*, 2014). Default parameters for count data normalization as recommended (Conesa *et al.*, 2016) were used to allow for control of log2 fold change shrinkage, custom p-value and fold change cut-offs. Genes were considered differentially expressed and retained for further analysis if the test statistics p-value (adjusted for false detection rate) (FDR) was less than 0.05 according to the method from Benjamini & Hochberg (1995). Heatmaps for visualizing the relationships between the different treatments were generated using the package Pheatmap v1.0.12 (Kolde, 2012) in R software (Team, R. C. 2017).

### 3.3.6 Experiment 8: Network construction

*Steps for the network construction*

The first step was to determine Pearson Correlation Coefficient (PCC) as shown in Figure 3.2.

Pearson Correlation Coefficient (PCC), which ranges from -1 (perfect negative linear coexpression) to +1 (perfect positive linear coexpression), is typically associated with

each edge to estimate the amount of coexpression between any gene pair, whereas 0 (no correlation) denotes the absence of any linear relationships between two genes.

where  $m$  is the number of samples,  $x_i$  is the vector holding all  $m$  expression values of gene  $g_i$ ,  $x_{k_i}$  denotes the expression of gene  $g_i$  in sample  $s$  and  $\bar{x}_i$  is the mean expression of gene  $g_i$  across all samples (figure 3.2)

The second step was to determine threshold: The soft threshold algorithm makes a gene expression network to be distributed with free-scale network by setting edges between any pair of genes whose correlation value exceeds this threshold by raising the absolute correlation value to a power  $\beta$ .

The third step was to construct a topological overlap matrix TOM that defines the topological overlap between two nodes based on the adjacency.

Where TOM is the Topological Overlap Matrix,  $a$  is the adjacency matrix,  $i$  is the row number of the Adjacency Matrix and TOM,  $j$  is the column number of the Adjacency Matrix and TOM.  $u$  increases from 1 to the maximum row number, also the maximum column number (Figure 3.2).

The adjacency matrix was transformed into a topological overlap matrix (TOM). A TOM matrix quantitatively describes the similarity in nodes by making comparison between two nodes and others.

### 3.3.6.1 Experiment 9: Graph filtering

The graph filtering technique adopted for this model was Maximal cliques. A Bron-Kerbosch-Algorithm was applied to find all possible cliques within a graph. A clique is a complete subgraph  $\subseteq G$ .

The process starts by determining a set  $C'$  of the maximal cliques  $C_i^{max}$  where a maximal clique is a complete subgraph  $C_i \subseteq G$  which is not a subset of another complete subgraph.

Whenever there is  $|C_i^{max}| > 1$ , where  $C_i^{max} \in C'$  a rating function  $r: C_i^{max} \rightarrow \mathbb{R}$ ,  $\forall C_i^{Max} \in C'$  is applied and maximal clique with the highest score selected. A key concept of networks is node connectivity, which measures the relative importance of the nodes in the network.

### **Node degree**

The *degree*,  $d_i$ , of gene  $i$  in a network of  $N$  genes, represents the number of nodes connected to gene  $i$ . Genes with large degrees are commonly referred to as hubs. Where  $a_{ij}$  is 1 if gene  $i$  is linked to gene  $j$  and 0 otherwise

This approach considers the number of nodes connected to gene  $I$ , edge Percolated Component (EPC) which assigns a random number between 0 and 1 to every edge and filters the graph by removing edges if their associated random numbers are less than the threshold and ecCentricity which considers distance between a vertex to all other vertices were used. The resulting features were compared with the maximal cliques based on the strength of the rules generated(support).

### **3.3.7 Experiment 10: Discretization**

Discretization is a data pre-processing step used in machine learning to transform continuous or numerical attributes into discrete ones (Hacibeyoğlu & Ibrahim, 2016). Features that did not have any connection in the network were filtered out and the remaining genes discretized using Equal Frequency Discretization (EFD). EFD is an unsupervised discretization method used in the absence of any knowledge of the class memberships of the instances. This method works by dividing a continuous attribute  $A=\{a_1, a_2, \dots, a_{n-1}, a_n\}$  into  $k$  intervals which include the same number of values. Each interval contains  $n/k$  bins where  $n$  is the number of values. This method was used because it reduces the effect of outliers and collects similar values in the same interval (Hacibeyoglu & Ibrahim, 2016). The discretization steps used are outlined below:

- Input:** The continuous values of attribute and number of intervals  $A = \{a_1, a_2, \dots, a_{n-1}, a_n\}$  and number of intervals  $k$ , where  $k > 0$ .
- Step 1:** Sort all values of in ascending order,
- Step 2:** Divide  $A$  by  $k$  intervals,
- Step 3:** Create bins according to number of elements in each interval,
- Step 4:** Determine boundaries of each interval by calculating the average value of the Maximum value of the current bin and the minimum value of the next bin,
- Step 5:** The continuous values of  $A$  are transformed into discrete ones by determining the interval that they belong to,
- Output:**  $A$  with discrete values

Two bins in step 3 were defined as described in Gallo *et al.*, (2016), with the final output being discretized measurements whereby a value of zero represented a gene that was under-expressed, while a value of one represented a gene that was overexpressed. The discretization process is summarized in Figure 3.3 below:

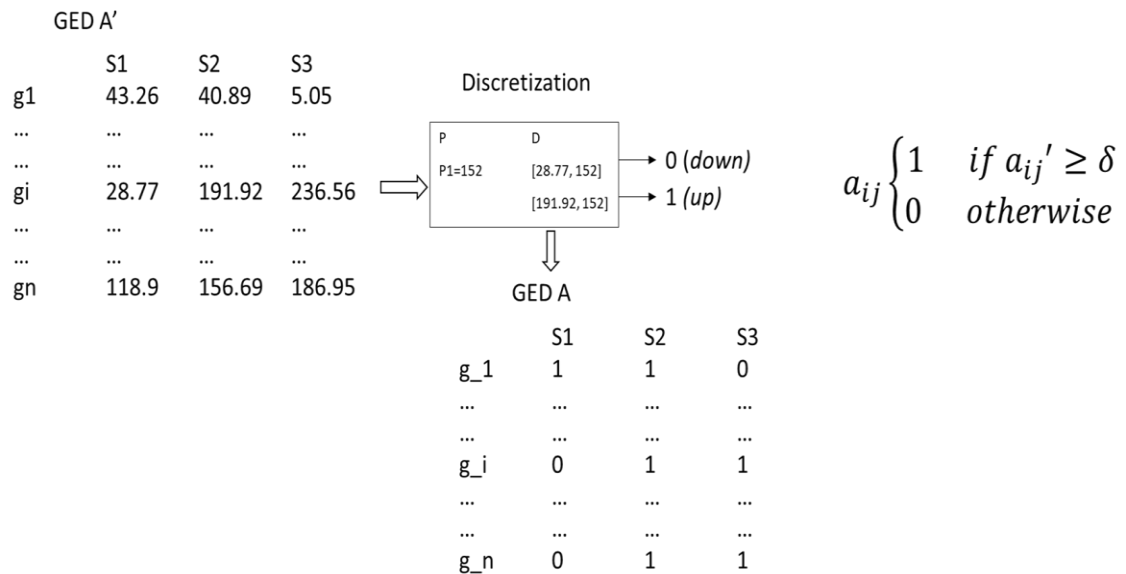


Figure 3.3: Discretization process

### 3.3.8 Experiment 11: Association rule mining

The steps followed in mining frequent item sets are described in the algorithm below:

---

## Apriori algorithm

---

```
Procedure Apriori (T, minSupport) {
//T is the database
// min_support is the minimum support
 $C_k$  : Candidate itemset of size k
 $L_k$  : frequent itemset of size k
 $L_1 = \{frequent\ items\}$ ;
For ( $k=2$ ;  $L_{k-1} \neq \emptyset$ ;  $k++$ ) {
 $C_k =$  Candidates-generated from  $L_{k-1}$ 
For each transaction t in database do {
 $L_k =$  candidates in  $C_k$  with minSupport
} //end for each
} //end for
Return  $\cup L_k$ ;
}
//Candidate generation
Candidates-generated ( $L_{k-1}$ )
 $C_k = \emptyset$ 
For all itemsets  $X \in L_{k-1}$  and  $Y \in L_{k-1}$  do
If  $X_1 \wedge \dots \wedge X_{k-2} = Y_{k-2} \wedge X_{k-1} < Y_{k-1}$  then
Begin
 $C = X_1, X_2 \dots X_{k-1} Y_{k-1}$ 
add C to  $C_k$ 
end
delete candidate itemsets in  $C_k$  whose subset is
not in  $L_{k-1}$ 
```

Figure 3.4: Apriori algorithm

The Apriori algorithm in figure 3.4 is explained as follows:

In the first step, the database  $T$  is scanned so as to calculate the support value for every item. In step two, items are stored as candidate itemsets with their support values. The next step is moving the candidates to the frequent itemsets if their support values are greater than or equal to  $\text{minsupp}$ . The frequent itemsets are joined to generate candidate itemsets where each candidate itemset is checked based on every sub-itemset which should be frequent itemset in the previous frequent itemsets. Therefore, the support value for each candidate itemset is calculated based on scanning the database in the first step. Each candidate itemset is checked based on every sub-itemset and should be frequent itemset in the previous frequent itemsets otherwise the candidate itemsets is deleted.

In association rule mining, a rule is typically described by three measures: support, confidence, and lift. These three represent the significance and interest of a rule. Support of a rule  $X \Rightarrow Y$  is equal to the support of the itemset  $X \cup Y$  and is defined as the probability of finding all the genes in sets  $X$ . Support of an itemset  $X$  is calculated as:

$$\text{Support}_D(x) = \frac{|\{T \in D \mid x \subseteq T\}|}{|D|} \quad (3.6)$$

The confidence of rule  $X \Rightarrow Y$  is the probability of finding all the differentially expressed genes in set  $Y$  as compared with the differentially expressed genes in set  $X$ . The confidence is calculated as:

$$\text{Confidence}_D(x \Rightarrow y) = \frac{\text{Supp}_D(X \cup Y)}{\text{supp}_D(X)} \quad (3.7)$$

Lift measures the strength of the rule and varies in the interval  $[0, \infty]$ . Lift is defined as:

$$\text{lift}(X \rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X) * \text{supp}(Y)} \quad (3.8)$$

Minimum support value between of 0.5 and 0.9, minimum confidence value of 0.9 and lift of  $\geq 2$  were the parameters used in generating rules.



The proposed Graph-based approach was compared to two other dimensionality reduction techniques which are Principal Component Analysis (PCA) and Recursive Feature Elimination (RFE).

### Principal Component Analysis (PCA)

Assuming dataset  $x^{(1)}x^{(2)}, \dots, x^m$  has inputs of  $n$  dimensions, this  $n$  – dimension data must be reduced to  $k$  – dimensional ( $k \ll n$ ) using PCA. The first step in PCA is standardization whereby the raw data should have unit variance and zero mean defined as:

$$x_j^i = \frac{x_j^i - \bar{x}_j}{\sigma_j} \forall j \quad (3.9)$$

In the second step, a covariance matrix of the raw data is calculated. The purpose of this step is to determine if there is any relationship between the variables in the input data set and how they differ from the mean in relation to one another. Variables can occasionally be highly connected to the point where they include redundant data. Therefore, a covariance matrix is computed in order to discover these relationships as shown in the equation below.

$$\Sigma = \frac{1}{m} \sum_i^m (x_i)(x_i)^T \Sigma \in R^{n \times n} \quad (3.10)$$

The third stage is calculation of eigenvector and eigenvalue of the co-variance. To find the primary components of the data, the linear algebra concepts of eigenvectors and eigenvalues must be computed from the covariance matrix using the equation below:

$$u^T \Sigma = \lambda, \quad (3.11)$$

$$U = [u_1, u_2, \dots, u_n], u_i \in R^n$$

In the fourth step, raw data is projected into a  $k$ -dimensional subspace, and this is followed by choosing the top  $k$  eigenvector of a co-variance matrix. The corresponding vector is calculated as shown in eqn. 3.12:

$$xi^{new} = \begin{bmatrix} u_1^T x^i \\ u_2^T x^i \\ \dots \dots \\ \dots \dots \\ u_k^T x^i \end{bmatrix} \in R^k \quad (3.12)$$

The raw data with  $n$  dimensionality is reduced to a new  $k$  dimensional representation. By eliminating the components with little information and using the remaining components as the new variables, principal component analysis allows dimensionality reduction without losing much information.

### **Recursive Feature Elimination**

The second feature selection approach used was Recursive Feature Elimination (RFE). This is a recursive process where features are ranked based on their importance (Rtayli & Enn, 2020). RFE employs machine learning models in computing features relevant scores. RFE first trains the model using all features and then computes the relevance score of every feature in the dataset. All the features with the least relevance score are ignored and this is followed by model retraining for computation of new relevant feature scores. This process is repeated until the final desired features are obtained (Rtayli & Enn, 2020).

### **3.4 Experiment 12: Classification**

Classification of the selected features from three techniques was performed using three classifiers namely Naïve Bayes, Sequential Minimal Optimization (SMO) and Multilayer Perceptron. The resulting classification accuracy of the different feature selection methods were compared. Features were the dependent variables while class was the independent

variable. In this case the expression levels (counts) of the features (genes) were used to predict the tissues type (diseased/normal).

### **3.4.1 Class balancing**

The two cancer datasets used in this study were highly imbalanced whereby the cancer samples were the majority class while normal samples were the minority class. Class imbalance is a problem that is typically encountered in disease-related datasets, such as cancer dataset used in this work. The majority class had a bigger number of cases than the minority class, which had a proportionally smaller number of occurrences. When using an imbalanced dataset, classifiers tend to prefer the majority class, resulting in very low classification rates for the minority class. It's also possible that the classifiers will classify all instances as belonging to the majority and ignore the minority. Therefore, for medical datasets a good sampling technique is essential. Various sampling strategies, such as undersampling, oversampling, and a combination of both, have been devised to address the problem of class imbalance. Through the removal of some data from the majority class (undersampling) or the addition of some artificially generated or replicated data to the minority class (oversampling), sampling procedures are presented to overcome class imbalance issue because data must be well-balanced to develop a solid prediction model from the training set (Kothandan, 2015).

Synthetic Minority Oversampling Technique (SMOTE) algorithm was used to balance the datasets. Oversampling of minority classes is done by creating synthetic samples to imitate minority classes and increase their number of instances in the training set (Rao, & Makkithaya, 2017). This approach has the advantage of preserving all the information from the original training dataset, as all observations from the majority and minority classes are kept. The number of instances ( $n$ ) and the nearest neighbors are two crucial parameters that are used to create these synthetic instances ( $k$ ). Overfitting is avoided because fresh minority instances are created by interpolating between numerous minority samples that are close together. Minority classes were increased based on the 5 k-nearest neighbors and defined  $n$  to equal classes.

### 3.4.2 Building classifier models

After the pre-processing phase which involved addressing class imbalance, 10-fold cross validation was applied to reduce the bias associated with random sampling of the training data. The original dataset was randomly partitioned into  $k$  equal size subsets in  $k$ -fold cross-validation. The categorization model was trained  $k$  times and tested while a single subset was kept as validation data for testing the model each time and the remaining  $k - 1$  subsets were used as training data. A Naïve Bayes (NB), multilayer perceptron (MLP), and Sequential Minimal Optimization (SMO) classification technique were chosen to In this experiment default parameters for MLP were used.

#### 3.4.2.1 Naïve Bayes algorithm

Naïve Bayes is efficient supervised learning method suitable for both binary and multiclass classification. The algorithm is based on Bayes' theorem (Bhavsar & Ganatra, 2012). Bayes theorem calculates the  $P(c|x)$ , posterior probability using  $P(x|c)$ ,  $P(c)$  and  $P(x)$ , as shown in the equation below.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \quad (3.13)$$

Where  $P(c|x)$  is the class posterior probability and  $P(c)$  is the prior probability class:

$P(x|c)$  is the likelihood that is the predictor given class probability.

$P(x)$  is the predictor prior probability.

#### 3.4.2.2 Sequential Minimal Optimization (SMO)

SMO is a supervised machine learning algorithm that belongs to the SVM classifiers (Hall *et al.*, 2009). The SVM algorithm works by building a hyperplane that separates different instances into their specific classes (Vapnik, 1998). Thereafter a pairwise a multiclass classification scheme is performed. Even when  $p > n$  SVM is functional without any alteration. SVM hyperplane is defined as:

### 3.4.2.3 Multilayer perceptron

Multilayer Perceptron (MLP) belongs to a class of feedforward artificial neural networks, which find complex patterns that a human programmer cannot extract by performing machine recognition. MLP has input layers (attributes), output layer (classes), and hidden layer(s) that are interlinked by various neurons. The optimization of interconnected weights is done by the backpropagation algorithm by training instances of the dataset (Tanwani *et al.*, 2009). In this experiment we used default parameters for MLP which are epochs = 500; learning rate of 0.3 and the momentum of updating weights was set to be 0.2. These are epochs = 500; learning rate of 0.3 and the momentum of updating weights was set to be 0.2.

### 3.4.2.4 Measures for performance evaluation

Classification accuracy, classification time, kappa statistic (KS), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE), were used to measure the model's performance for every classifier before and after feature selection using the various approaches. The working principle of the techniques is described below.

#### Classification accuracy

Accuracy measures how well a test can predict different categories. It shows the number of samples that are correctly classified into their respective classes. Accuracy is expressed as a percentage which is calculated by formula shown in equation 3.13:

$$Accuracy = \frac{TN+TP}{TN+TP+FN+FP} 100\% \quad (3.14)$$

Where:

- TN is the **True Negative**: samples classified as negative class but belongs to negative class.

- TP is the **True Positive**: samples classified as positive class and they belong to positive class
- FN is the **False Negative**: samples that belong to a positive class but have been classified in a negative class.
- FP is the **False Positive**: samples classified as positive, but they belong to the negative class.

### **Classification Time**

This is the total CPU time that is required to build a classification model as well as the training time required to predict the output of the test data.

### **Kappa Statistic (KS)**

Kappa statistic is calculated to evaluate measurement accuracy. The closer the  $K$  value is from 0 to 1, the more reliable the classification. When  $K$  equals 1, the correctness of classification is the safest. On the other hand, when  $K$  equals 0, the chance of classification is right and unreliable. Calculation of Kappa value is given in equation 3.14.

$$K = \frac{P(A) - P(E)}{1 - P(E)} \quad (3.15)$$

Where:

$K$ : is the Kappa Statistics

$P(A)$ : is the Percentage of agreement

$P(E)$ : is the Agreement chance

### **Mean Absolute Error (MAE):**

MAE is useful measurement to use for performance evaluation of algorithm. As given in (3.15), it is calculated by taking the average of all absolute errors (Chai & Draxler, 2014).

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - x| \quad (3.16)$$

Where

n= the number of errors

$\Sigma$  = summation symbol (which means “add them all up”)

$|x_i - x|$ = the absolute errors

### **Root Mean Squared Error (RMSE)**

RMSE is popular measurement for performance evaluation measurement which calculates the error without canceling the positive and negative error (Margaret &Sridhar, 2006).

$$RMSE = \sqrt{\frac{1}{n} \left[ \sum_1^N (Q_{exp} - Q_{cal})^2 \right]} \quad (3.17)$$

Where

n = the sample size

$(Q_{exp} - Q_{cal})^2$  = the difference squared

After classification performance measure we then did a 70: 30 subsets on data and performed 10-fold cross validation of the accuracy and then recorded the accuracy and F-measure. This was repeated to 20 times across each feature selection method followed by Kruskal Wallis H-statistic to test if there was significant difference in the mean ranks of the groups.

## CHAPTER FOUR

### RESULTS AND DISCUSSION

In this section the results and their discussions for this study are presented based on the study objectives.

#### **4.1 Data preprocessing results**

##### **4.1.1 Trimming low quality reads results**

Raw reads per sample in the Tsetse fly antennae dataset ranged between 23 and 73 million. A subset of 3 million reads from each sample was used to evaluate three different trimming algorithms. Quality-based trimming of RNA-Seq data is usually done to improve mapping of reads to the reference genome (Del Fabbro *et al.*, 2013). The three algorithms displayed different patterns in terms of the surviving paired reads and the alignment rate. The window-based algorithm performed better than the other two approaches whereby the percentage of the surviving reads ranged between 83.39% and 90.87% (Figure 4.1).



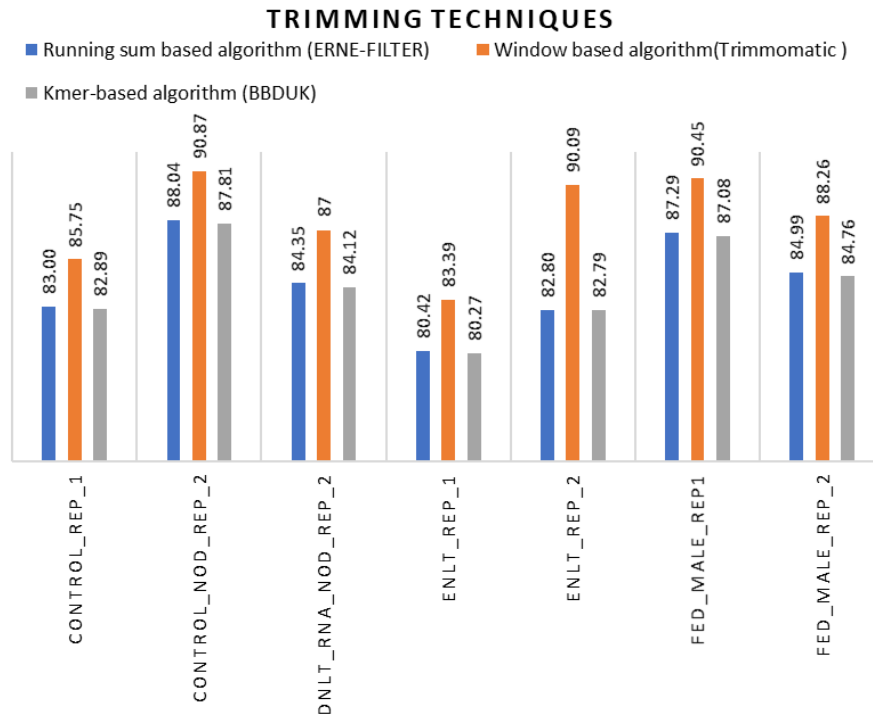


Figure 4.1: **Comparison of the output from three trimming algorithms.** The values show the percentage of surviving paired end reads for each sample.

The second algorithm evaluated in this study was BBDUK which stands for Decontamination using Kmers. The number of surviving reads after running this algorithm ranged between 80.27% and 87.81% across all samples (Figure 4.1). The third approach evaluated in this study was a running sum algorithm which works by adding the sum of the values from the left side of the array and checks the reads that meets the minimum quality score. This algorithm had surviving paired end reads ranging between 80.42% and 88.26% across all samples (Figure 4.1). In this study, the window-based algorithm performed better than *K-mer* based and running sum approaches. Window-based algorithms have a user-defined window that spans the read from 5' to 3'. Therefore, given a window of length  $l$  and a quality threshold  $Q$ , the algorithm cuts the 3'-end when the average quality drops below  $Q$ . The main advantage of this approach is that it handles paired end reads and this may be responsible for the good performance. In addition, the performance of this algorithm is improved by calculating the average since any values

with high quality score can increase the average of the values with the lower scores. On the other hand, BBDUK which is a Kmer-based technique which constructs a truncated suffix tree to a depth equal to the pattern size to be searched and works by comparing substrings of length  $K$  while discarding unmatching characters. The challenge associated with this approach is that larger k-mers have a higher risk of not having outward vertices from every k-mer. This is due to larger k-mers increasing the risk that it will not overlap with another k-mer by  $k - 1$ . Running sum algorithm works by adding the sum of the values from the left side of the array and checks the reads that meets the minimum quality score and deletes the reads that is lower than the user defined score. Therefore, the window-based algorithm performed better than the other two algorithms. Williams *et al.*, (2016) reported that reads trimmed with a window-based algorithm mapped better due to less aggressive trimming. Another confirmation is by He *et al.*, (2020) who reported a survival frequency of between 97% and 98%.

#### **4.1.2 Indexing and mapping**

The output of the best trimming technique (Window-based algorithm – Trimmomatic) was used as the input for the indexing and mapping steps. In this phase, three indexing/mapping techniques were evaluated. These mapping tools were classified based on the data structure they use i.e., Burrow's Wheeler transform-based techniques, Hash table-based techniques and Suffix array-based techniques. Results showed that BWA performed better than Bowtie2 in terms of the alignment across all the samples with accuracy values between 93% and 97.97%. Bowtie2 accuracy was between 83.81% and 90.01% (Figure 4.2). The third algorithm in this category was Spliced Transcripts Alignment to a Reference (STAR) algorithm which uses a suffix array to provide faster processing. Accuracy values ranged between 83.32% and 95.04%. According to the results, SMALT mapped with higher accuracy than NGM with the accuracy ranging between 93.27% and 97.89% (Figure 4.2).

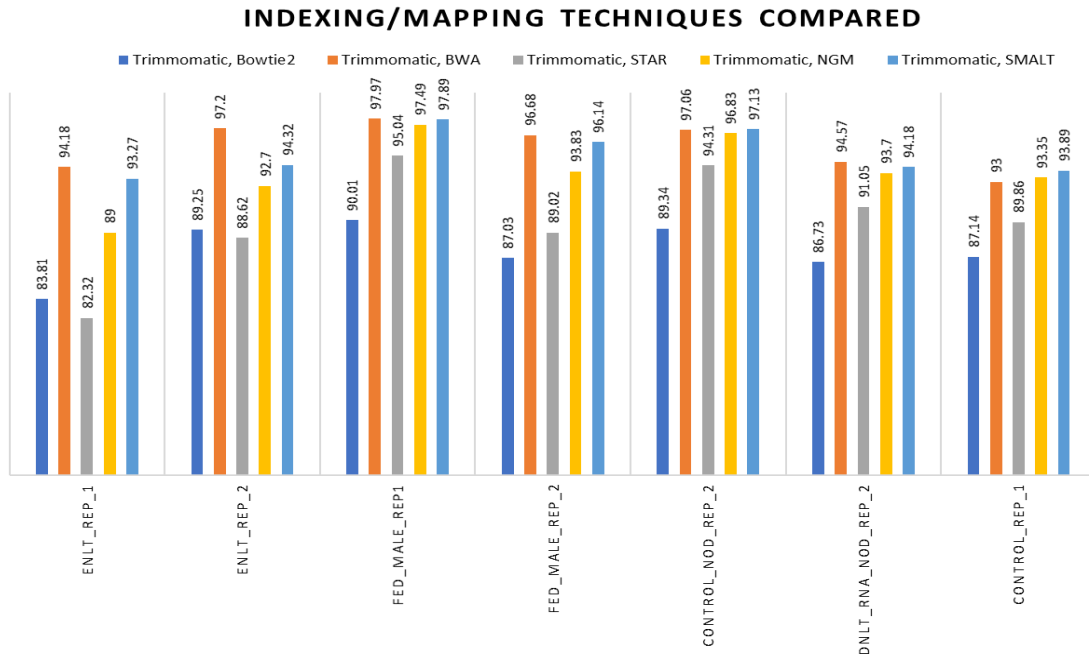


Figure 4.2: **Comparison of the various indexing and mapping algorithms.** The values indicate the percentage of correctly mapped reads per sample.

BWA's performance can be attributed to its working principle. BWA algorithm works by constructing a suffix array and Burrows-Wheeler-Transformation (BWT), and subsequent matching of the sequences is done using a backward search. During the genome indexing phase, BWA combines Burrows-Wheeler transform and an FM-Index that enables it to achieve a linear lookup time for an exact match. This facilitates efficient search at a goal state and works backward until the initial state is encountered. This function facilitates in getting more matching *k-mers* thus higher accuracy. Bowtie2 also indexes the genome using FM-Index. To search and find the match in the reference genome, Bowtie2 performs a Depth-first search on the prefix *trie* and stops when the first qualified hit is found.

The highly efficient mapping STAR algorithm is due to the two-step process which is Seed searching and Clustering, stitching, and scoring. For every read that STAR aligns, STAR will search for the longest sequence that exactly matches one or more locations on the reference genome. These longest matching sequences are called the Maximal

Mappable Prefixes (MMPs): In the second phase, STAR builds alignments of the entire read by stitching together all the seeds that were aligned to the genome in the first phase. First, the seeds are clustered together by proximity to a selected set of ‘anchor’ seeds. All the seeds that map within user-defined genomic windows around the anchors are stitched together assuming a local linear transcription model. The size of the genomic windows determines the maximum intron size for the spliced alignments. If an alignment within one genomic window does not cover the entire read sequence, STAR will try to find two or more windows that cover the entire read, resulting in a chimeric alignment, with different parts of the read mapping to distal genomic loci, or different chromosomes, or different strands.

Both HISAT2 and STAR have been reported to obtain greater coverage values for alignments >1,000 bases, implying that these two tools are better at mapping bigger transcripts than the other aligners examined (Musich *et al.*, 2021). Because of the growing volume of data produced by high-throughput sequencers, alignment speed may be a consideration when choosing a mapper. In all the testing, the STAR aligner was the fastest i.e., 25.4 times quicker than Bowtie2 and 86 times faster than BWA. Mapping speed of STAR aligner was also reported in earlier study by Ziemann *et al.*, (2016) where STAR was 15 percent faster when used on human data. In general, longer reads took longer to process, with Bowtie2 experiencing the highest lag (46x) as compared to BWA mem (5.5x) and these findings suggested that STAR could be a viable high-throughput option for researchers (Ziemann *et al.*, 2016). The main downside of this algorithm is the memory requirement of around 32 GB as compared to methods that use Burrows-Wheeler transform.

SMALT and NextGenMap (NGM) are the two algorithms that use a hash table data structure. SMALT algorithm uses hash index of short words of less than 15 nucleotides long. From every read, potential matching segments in the identified seed matches in the index and afterward aligned with the read using a banded Smith-Waterman (SW) algorithm. The SW algorithm consists of a matrix-filling phase and a back tracing phase. The matrix-filling phase computes the similarity scores of the arbitrary regions of

sequences, and the back tracing phase identifies the local alignments that can be found from the highest-scoring matrix cell. Given the two sequences of lengths  $m$  and  $n$  ( $\geq m$ ), the time complexity of the SW algorithm is  $O(mn)$ . NextGenMap supports a Smith-Waterman (SW) and a Needleman-Wunsch (NW) banded alignment computation, thus allowing for a user-defined maximal number of admissible consecutive insertions and deletions. A lookup table is computed that assigns putative genomic positions to a read based on a short exact matching word. Preprocessing of the read counts that have mapped to a reference genome is an important step that eliminates non-differentially expressed features. Features with a False discovery rate (FDR)  $< 0.05$  are usually considered as being of biological significance (Von Der Weid *et al.*, 2015). In this study the filtering step eliminated 18.9% low-count features from subsequent analysis.

#### 4.1.3 Feature counting results

After the evaluation of feature extraction techniques, the optimal approaches were used to preprocess the entire dataset. The resulting reads ranged between 66,498 and 22,749 (Figure 4.3)

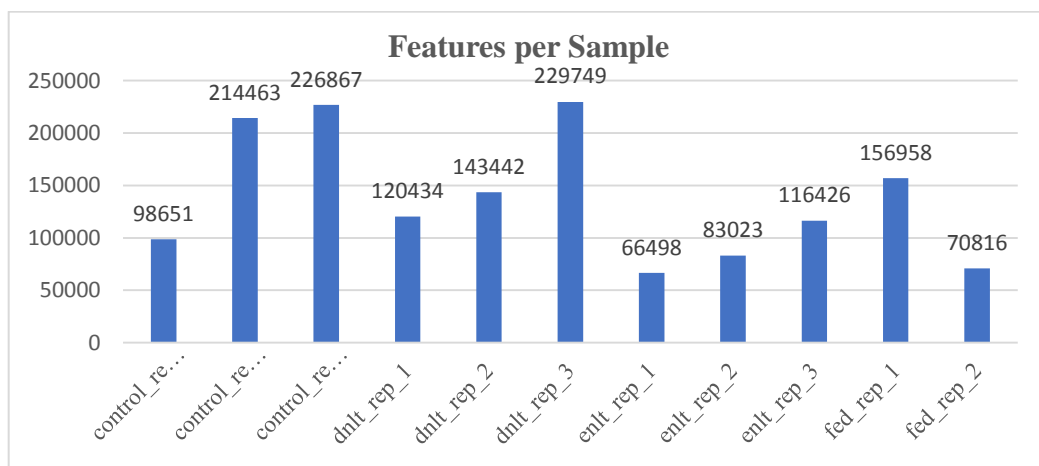


Figure 4.3: **Final feature counts after feature extraction.**

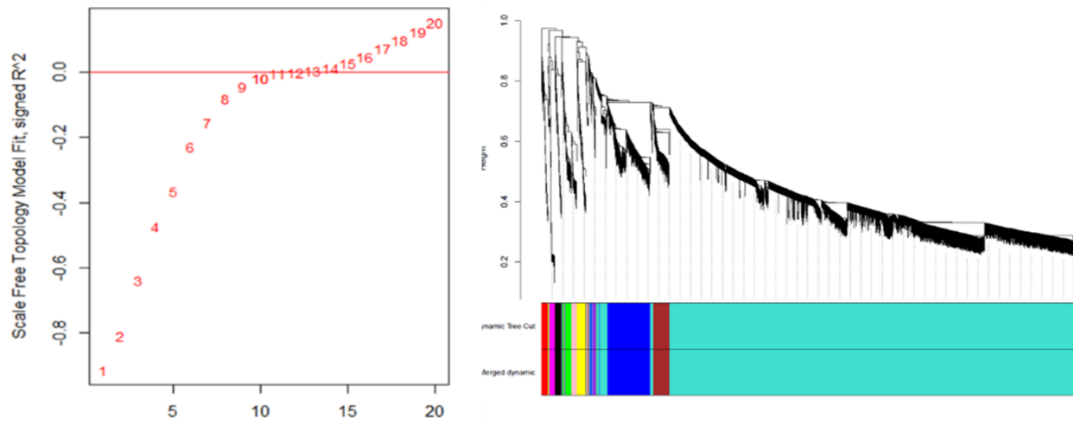
#### **4.1.4 Normalization and differential expression analysis results**

In the previous phase, a subset of only 3Million reads were used for feature extraction. However, for the development of the graph-based feature selection model for phenotype prediction, the entire dataset with raw reads per sample ranging between 23 and 73 million was used. Preprocessing involved elimination of non-differentially expressed features and normalization. During differential gene (feature) expression analysis, 2,097 low-count features were filtered out from the total set of 11,089 features leaving a final tally of 10,921 features. Features with a False discovery rate (FDR)  $< 0.05$  are usually considered as being of biological significance (Von Der Weid *et al.*, 2015).

## **4.2 Graph construction**

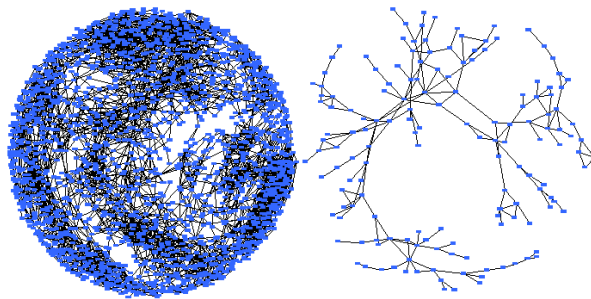
### **4.2.1 Graph threshold and module detection results**

A scale-free topology weighted gene expression network was constructed using WGCNA based on a soft thresholding power ( $\beta$ ). From candidate powers of between 1-20,  $\beta=12$  returned a scale-free topology fit index and an adjacency matrix based on the criterion of approximate scale-free topology as shown in Figure 4.4a. Using the dynamic tree cutting algorithm, all the features were grouped into modules as shown in Figure 4.4b. The dataset had 12 modules, which ranged in size from 42 to 9325 genes per module. A global network was generated (Figure 4.4c). This network had 2,110 nodes and 4783 edges but a very low network density of 0.002 as shown in Figure 4.4d.



**a**

**b**



**c**

Summary Statistics	
Number of nodes	2110
Number of edges	4783
Avg. number of neighbors	4.689
Network diameter	50
Network radius	25
Characteristic path length	14.632
Clustering coefficient	0.283
Network density	0.002
Network heterogeneity	0.773
Network centralization	0.011
Connected components	2
Analysis time (sec)	0.844

**d**

Figure 4.4: a) **Scale-free fit index versus soft-thresholding power**, b) **Grouping of features into modules based on the expression patterns**, c) **Global network for all features**, d) **Global network statistics**

#### 4.2.1.1 Correlation between modules

Potential relationships between features in the turquoise, blue, brown, and yellow modules were explored by visualizing only the functionally annotated features in the form of a network (Figure 4.5a). Filtering was done to reduce the size of the network by excluding all the features with a degree value of less than 5. The final network had 51 nodes and 148 edges (Figure 4.5b).

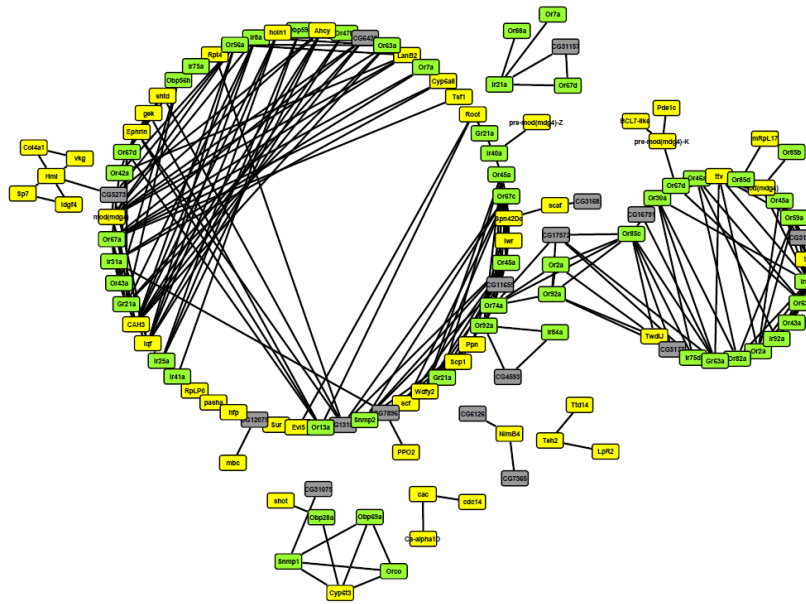


Figure 4.5a

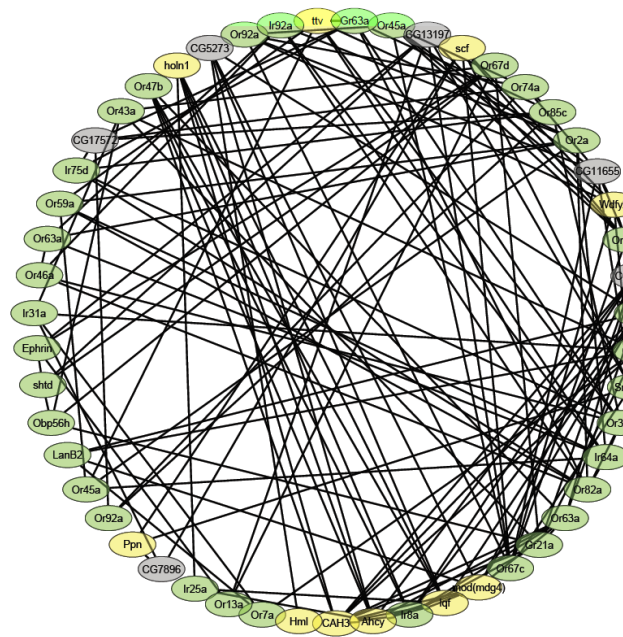
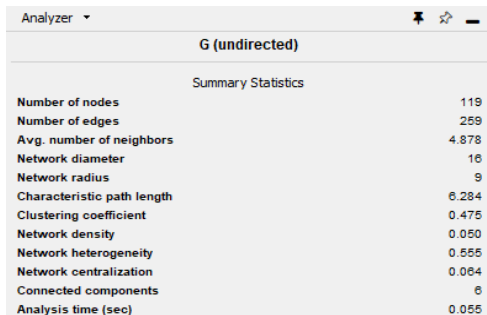


Figure 4.5b

Fig. 4.5: a) Co-expression networks for the genes in the turquoise, blue, brown and yellow modules. b) Filtered co-expression network for the genes with a degree value

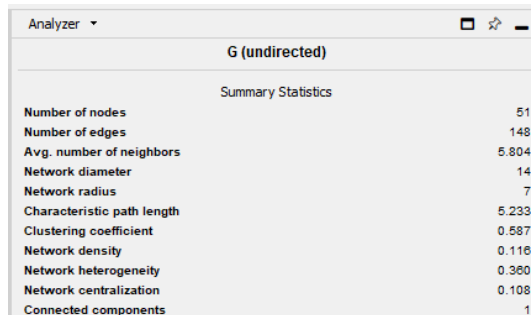


greater than five. Chemosensory genes are depicted in green, non-chemosensory in yellow and those with unknown function in grey (Figure 4.5a and 4.5b).



G (undirected)	
Summary Statistics	
Number of nodes	119
Number of edges	259
Avg. number of neighbors	4.878
Network diameter	16
Network radius	9
Characteristic path length	6.284
Clustering coefficient	0.475
Network density	0.050
Network heterogeneity	0.555
Network centralization	0.064
Connected components	6
Analysis time (sec)	0.055

**Figure 4.5c**



G (undirected)	
Summary Statistics	
Number of nodes	51
Number of edges	148
Avg. number of neighbors	5.804
Network diameter	14
Network radius	7
Characteristic path length	5.233
Clustering coefficient	0.587
Network density	0.116
Network heterogeneity	0.360
Network centralization	0.108
Connected components	1

**Figure 4.5d**

**4.5c) Network summary statistics before filtering.** 4.5d) Summary statistics after filtering.

Filtering improved density of the network from 0.05 to 0.116 while overall clustering coefficient improved from 0.475 to 0.587 (Figure 4.5c and 4.5d). The degree, average shortest path length and clustering coefficient for the top genes with a node greater than 8 are shown in **Table 4.1**.

**Table 4.1: Network topology for the top genes with a node degree greater than 8.**

No.	Gene Symbol	Clustering Coefficient	Degree	Betweenness Centrality
1	<i>Ahcy</i>	0.47	11	0.06
2	<i>CAH3</i>	0.53	11	0.07
3	<i>Ir64a</i>	0.38	10	0.1
4	<i>Or67c</i>	0.44	10	0.2
5	<i>Ir8a</i>	0.6	10	0.03
6	<i>Or67a</i>	0.29	10	0.08
7	<i>lqf</i>	0.56	9	0.02
8	<i>mod(mdg4)</i>	0.58	9	0.01
9	<i>Or30a</i>	0.31	9	0.24
10	<i>Or45a</i>	0.5	9	0.11
11	<i>Or85c</i>	0.36	9	0.32
12	<i>CG11655</i>	0.57	8	0.18
13	<i>Wdfy2</i>	0.61	8	0.03
14	<i>Or82a</i>	0.39	8	0.05
15	<i>Or2a</i>	0.54	8	0
16	<i>Or67d</i>	0.32	8	0.22
17	<i>Or45a</i>	0.46	8	0.2
18	<i>Or63a</i>	0.68	8	0.02
19	<i>Or56a</i>	0.46	8	0.09
20	<i>Gr21a</i>	0.43	8	0.14

Genes *CAH3*, *Ahcy*, *Ir64a*, *Or67c*, *Ir8a* and *Or67a* can be regarded as the top hub genes since they had degree values above 10. Fourteen of the top 20 genes are associated with chemosensation, which is an important biological function in insects. Clustering coefficient of hub gene nodes ranged between 0.29 and 0.68 and this is an indication that some parts of the network were more intricately connected than others. Potential relationships between features were explored by visualizing them in the form of a network. Weighted gene co-expression network analysis is a statistical model that is used on biological data and works even on non-model organisms (Degli Esposti *et al.*, 2019). Therefore, when WGCNA was used on *G. m. morsitans* data, the dimensionality of the data was reduced, and meaningful patterns could be extracted using network centrality measures. The resulting co-expression network was useful in selecting genes with significant connectivity patterns that are biologically meaningful. Node degrees helped identify two genes (*CAH3* and *Ahcy*) that had a degree of 11 and therefore these can be

regarded as the top hub genes. In *Drosophila melanogaster*, CAH3 is a carbonate dehydratase involved in generation of protons and bicarbonate from carbonic acid (Overend *et al.*, 2016). Ahcy is involved in methionine biosynthesis and metabolism (Brosnan & Brosnan, 2006). The observed betweenness centrality measures mean that some of the genes such as Teh2 and Ir21a which had values of 1 and 0.8 respectively would have more control over the network as compared to those with lower values. Therefore genes/nodes with high betweenness centrality values are more biologically informative in a module (Riquelme & Lubovac-Pilav, 2016). Closeness centrality measures for majority of the nodes were between 0.1 and 1, which means the resulting network was closely connected, while twenty-seven of the nodes in the network had a clustering coefficient of 1 an indication of complete node connection (Liu, 2018).

### **4.3 Discretization results**

A set of 308 features identified as differentially expressed after exposure to either repellent ( $\delta$ -nonalactone) or attractant ( $\epsilon$ -nonalactone) was further filtered to 180 features by the graph-based approach, 181 by PCA and 201 features by RFE. All these features were discretized for further analysis. A sample results of the discretized and transformed data is presented in a transaction format where the samples represent transaction IDs and features represent items (Figure 4.6).

Estimate Missing Values		Transform data: log2 logN log10 z-score add = <input type="text" value="0"/> mult = <input type="text" value="1"/>				Discretize	
norm_deseq1.csv [Steady State]							
Genes	CTRLP_R1	CTRLP_NO...	CTRLP_R3	DNLT_R1	DNLT_NOD...	DNLT_R3	ENLT_MAL...
GMOY000368-RA	1	0	1	1	0	0	1
GMOY000369-RA	1	0	0	0	0	0	1
GMOY000372-RA	0	0	0	1	1	1	0
GMOY000373-RA	0	0	1	1	0	0	1
GMOY000374-RA	0	0	1	0	1	0	0
GMOY000375-RA	0	1	1	0	1	0	0
GMOY000376-RA	0	1	0	0	0	0	0
GMOY000377-RA	0	0	0	0	0	0	1
GMOY000378-RA	0	1	0	0	0	0	0
GMOY000379-RA	0	0	1	0	0	1	0
GMOY000380-RA	0	1	0	1	1	1	1
GMOY000381-RA	1	1	1	1	1	0	0
GMOY000382-RA	1	0	0	1	1	0	1
GMOY000383-RA	0	1	1	1	1	0	0
GMOY000384-RA	1	0	0	1	0	0	1

Figure 4.6: A sample output of the discretization process. Count data is converted from continuous to discrete format.

In this study, two states were considered during discretization where the value of zero represents no expression or not present and the value of 1 represents expression or present (Figure 4.6). Discretization is a dimensionality reduction technique which converts continuous values to discretized values. Another popular method is to employ a ternary set of discretization symbols, which are  $\{-1, 1, 0\}$ , representing downregulation, upregulation, or no change at all. Nonetheless, the values in matrix  $A'$  can be discretized to an arbitrary number of symbols using a multilevel approach. The inference technique that relies on the discretized data determines the 'degree of discretization.' However, because the loss of information diminishes as  $k$  grows in value, the computing complexity of the inference algorithm increases, the trade-off between loss of information and computational complexity may also play a role in determining the 'degree of discretization' (Gallo *et al.*, 2016)

The discretized data is crucial for association rule mining especially when dealing with gene expression values which are continuous. Discretization facilitates generation of informative rules as well as reduction of computation resources such as memory during

rules generation. When performing discretization, biological and statistical condition must be met by discretization methods. The expression values must be divided into two classes that resemble lowly and highly expressed genes. The other condition is that discretization technique should ensure a sufficient distribution of genes across classes with minimal information loss (Lauria *et al.*, 2020).

#### **4.4 Apriori Algorithm-Based Association rule analysis**

At a minimum support of 0.5 and a confidence of 0.99, only the features generated by a graph-based approach generated 801 rules. Features selected PCA and RFE feature selection approaches for this data provided zero rules at the provided minimum support.. These rules were further filtered using a lift of  $\geq 2$  to empirically retain only the highly dependent rules as the best results. Lift values lower than 2 or support values less than 0.5 results into many redundant rules, however a support value greater than 0.5 resulted to no rules as per the *Glossina* dataset. Genes with no assigned biological function were of interest since the objective was to find out if association rule mining could help predict their phenotype. The analysis therefore narrowed down to the rules that implied an association between known genes and those with no known function. Twenty-two representative rules are shown in Table 4.3.

**Table 4.2: Association rules among genes that showed significant upregulation after exposure to an attractant ( $\epsilon$ -nonalactone).**

Set A shows association between genes with unknown function and chemosensation genes; Set B shows association between genes with unknown biological function and non-chemosensation genes ( $\epsilon$ -nonalactone).

No.	Association rule	Set
1.	{Ir84a, Or2a, Or42a, Or56a} => {CG3679}	
2.	{Or2a, Or42a, Or56a, Or49b} => {CG3679}	
3.	{Ir84a, Or42a, Or56a, Or49b} => {CG3679}	
4.	{Or88a, Gr63a, CG5273, CG17572} => {CG18480}	
5.	{CG4950, Or88a, Gr63a, CG5273} => {CG18480}	
6.	{Or88a, Gr63a, CG17572, CG31663} => {CG18480}	
7.	{Or88a, Gr63a, CG5273, CG17572} => {CG31663}	A
8.	{CG4950, Or88a, Gr63a, CG5273} => {CG31663}	
9.	{CG4950, Or88a, CG18480, Gr63a} => {CG31663}	
10.	{CG4950, Or88a, Gr63a, CG31663} => {CG17572}	
11.	{Or88a, Gr63a, CG5273, CG31663} => {CG17572}	
12.	{Or88a, Gr63a, CG17572, CG31663} => {CG5273}	
13.	{CG4950, Or88a, Gr63a, CG17572} => {CG5273}	
14.	{CG4950, Or88a, CG18480, Gr63a} => {CG5273}	
15.	{Tsf1, Scp1, vkg, Adgf.A} => {CG6126}	B
16.	{Ppn, Sp7, NtR, Adgf.A} => {CG6126}	
17.	{vkg, Sp7, NtR, Adgf.A} => {CG6126}	
18.	{LanB2, Sp7, NtR, Adgf.A} => {CG6126}	
19.	{Sp7, NtR, Adgf.A, CG6126} => {CG3168}	
20.	{Ppn, NtR, Adgf.A, CG6126} => {CG3168}	
21.	{Idgf4, NtR, Adgf.A, CG6126} => {CG3168}	
22.	{Ppn, Sp7, Adgf.A, CG6126} => {CG3168}	

Association rules are represented as  $X \Rightarrow Y$ , where X and Y are items contained within a dataset/database and  $X \cap Y = \emptyset$ . X is the antecedent and Y is the (Table 4.3). This rule means that whenever X which is antecedent is present even Y which is consequent will be present. Support indicates the frequency of the itemset appearance in the dataset and the confidence indicates how often a rule has been found to be true. Support value of 0.5 means 50% of the items (genes) are found in the transaction and 90% of the rule are true (Confidence). The lower support means that most of the items are not frequently found together. The lift value is used to measure the rule importance. A lift of greater than 2 achieved by graph-based feature selection approach indicates the degree to which any two

occurrences depend on each other and this is an indication that those rules are useful in consequent prediction. Association rule mining enabled identification of itemset patterns based on the RNAseq genes expression patterns. The first 22 rules indicate that there is a relationship among the genes (itemsets) expression in each condition (transaction) with the following genes CG18480, CG31663, Ir84a, CG17572, CG5273, Gr63a, Or88a, Or49b, Or2a, Or56a, Or42a, CG3679, Adgf-A, NtR, teq, NimB4, Scp1, CG3168, CG6126, Ppn, vkg, LanB2, Tsf1, scf, Sp7 and Idgf4 always being up or downregulated in response to either repellent ( $\delta$ -nonalactone) or attractant ( $\epsilon$ -nonalactone). The identified rules in this study are biologically significant based on the concept that similar items as in market basket analysis appear together is clearly shown in the results. For example, where the genes CG18480, CG31663, CG17572, CG5273 and CG3679 referred to as consequents (right side) were up (highly expressed), all the genes on the rule antecedent (left side) were also up. The rest of the rules can be interpreted in a similar manner. Genes Ir84a, Gr63a, Or88a, Or49b, Or2a, Or56a and Or42a are involved in chemosensation in insects. However, genes CG18480, CG31663, CG17572, CG5273 and CG3679 that are co-expressed with the chemosensation genes have no assigned biological function. Therefore, it is possible that these genes also play a role in chemosensation due to their co-expression and association with chemosensory genes. Only two genes (CG3168 and CG6126) were associated with the top non-chemosensory genes that were upregulated due to exposure to an attractant. A lift value greater than 2 in generating the rules because values greater than 1 indicate that consequent and antecedent are dependent on one another (Ahmadon & Yamaguchi, 2018).

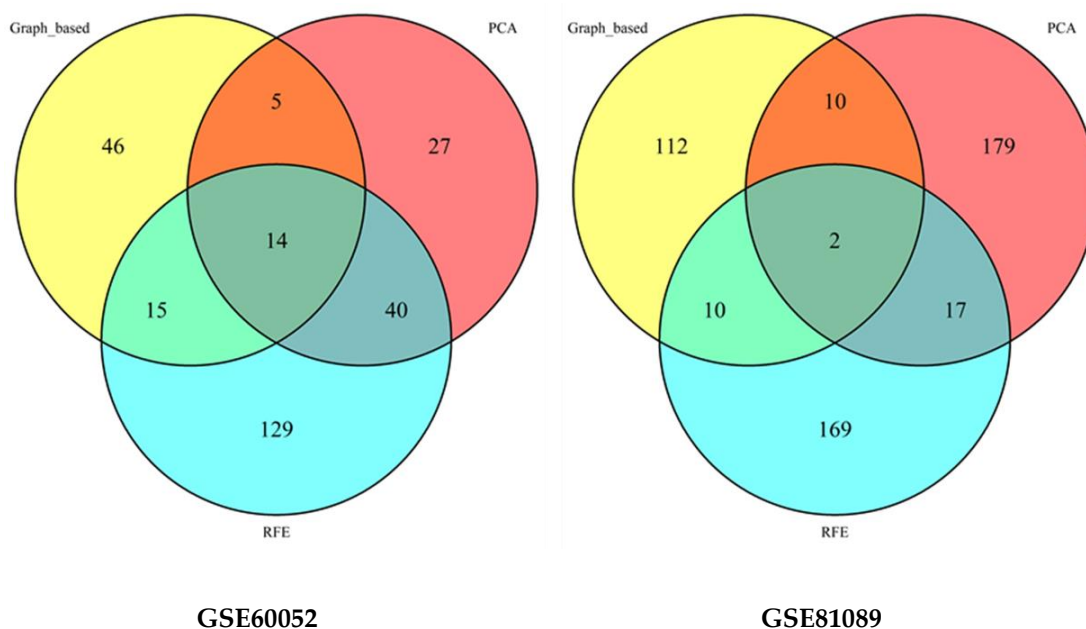
#### **4.5 Model validation results**

To validate the model, two cancer datasets described in the methodology were used. Both datasets had 28,089 initial features as summarized in Table 4.3. Preprocessing involved elimination of non-differentially expressed genes and normalization which resulted in 12.2% features for small-cell lung cancer and 43.2% features for non-small-cell lung cancer after preprocessing. Thereafter, three feature-selection approaches were used to further filter the features to retain only informative features. RFE retained the highest

number of features in both datasets followed by PCA as shown in Table 4.3 and Figure 4.7.

**Table 4.3: Output from normalization and feature selection using PCA, RFE and graph-based approaches.**

Dataset	Number of Features	Feature Selection method			
		Preprocessed	Graph	PCA	RFE
GSE60052	28,089	3423 (12.2%)	80	86	198
GSE81089	28,089	12,145 (43.2%)	134	208	270

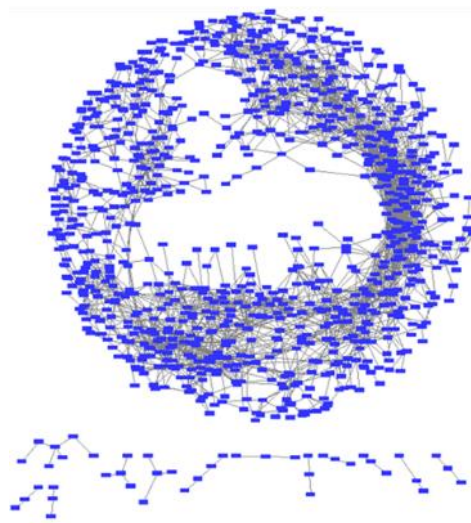


**Figure 4.7: Features selected by each of the methods from the two datasets.**

A graph-based feature-selection approach retained 80 for the SCLC dataset and 134 features for the NSCLC dataset. Among the three feature selection methods, 14 similar features were picked for GSE60052 dataset and only two similar features were picked by all feature selection methods as shown in figure 4.7. features selected by the graph are fewer and this can be attributed to its working principle where it only considers the



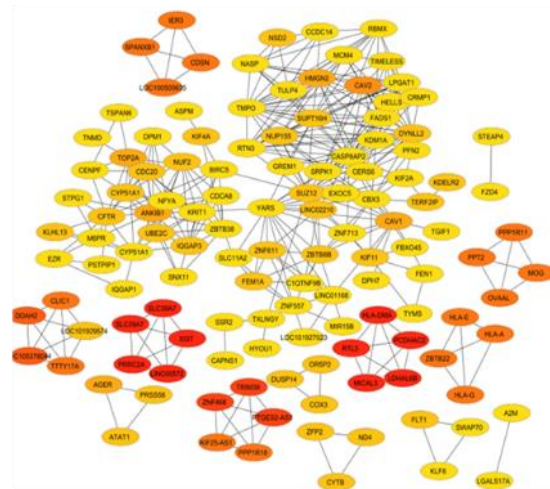
connecting features. The other reason is the extra filtering step using maximal cliques which is not done by other feature selection methods. This leads to retention of only the features that have the highest maximal clique score. Figure 4.8 presents the networks for the two datasets before and after filtering using maximal cliques. The networks were filtered using MCC technique to obtain a reduced network. Filtering changed the network density as well as the overall clustering coefficient in both networks as shown in Figure 4.9 c and d.



**a**

Summary Statistics	
Number of nodes	990
Number of edges	3154
Avg. number of neighbors	6.596
Network diameter	33
Network radius	17
Characteristic path length	8.618
Clustering coefficient	0.318
Network density	0.007
Network heterogeneity	0.873
Network centralization	0.030
Connected components	13
Analysis time (sec)	0.428

**c**

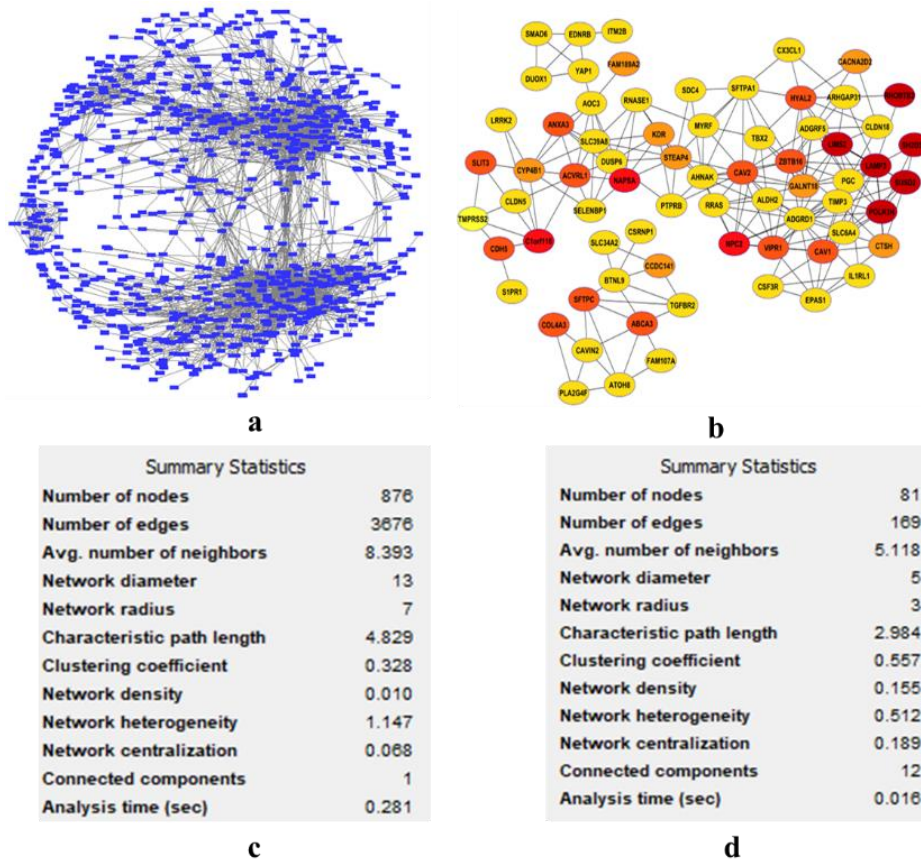


**b**

Summary Statistics	
Number of nodes	134
Number of edges	396
Avg. number of neighbors	7.725
Network diameter	7
Network radius	4
Characteristic path length	3.183
Clustering coefficient	0.553
Network density	0.098
Network heterogeneity	0.670
Network centralization	0.211
Connected components	15
Analysis time (sec)	0.016

**d**

Dataset GSE81089



Dataset GSE81089

Figure 4.8: **Network diagrams for the 2 datasets:** a) network before filtering while, b) networks after filtering with maximal clique. On the filtered networks, different colors denote expression levels with red color showing features that were highly expressed.

#### 4.6 Association Rule Mining

Features from PCA, RFE and graph-based selection methods were discretized and analyzed to find possible associations using Apriori. The resulting number of rules, maximum confidence, support, and lift values are summarized in Table 4.5.

**Table 4.4: Rules generated using Apriori from features selected using three different approaches**

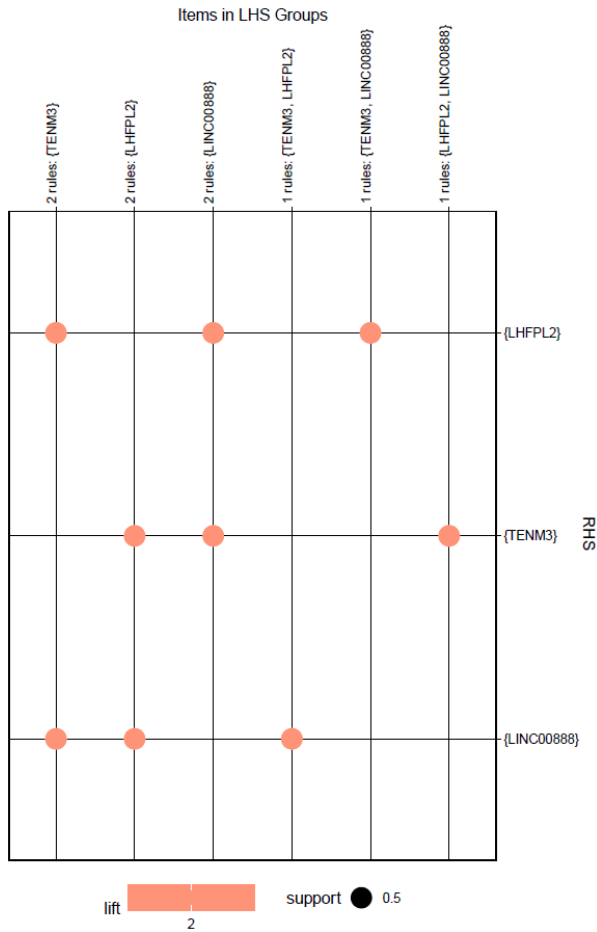
Dataset	Selection Method	Support	Confidence	Lift	No. of Rules	Non-Redundant Rules
GSE60052	Graph-based	0.5	0.9	2	19	15
	PCA	0.4	0.9	2	38	38
	RFE	0.3	0.9	1.98	357,986	112,357
GSE81089	Graph-based	0.5	0.9	2	36	36
	PCA	0.4	0.9	1	121	121
	RFE	0.4	0.9	1	899	884

As shown in Table 4.4, a graph-based feature-selection approach gave 15 and 36 non-redundant rules, respectively, from the two datasets at a support of 0.5 confidence value of 0.9 and a lift of 2. PCA and RFE feature-selection methods did not generate any rules at a support of 0.5. Features selected by RFE had the lowest maximum support and lift, and this led to the generation of too many redundant rules. The lower support as provided by RFE feature selection approach implies that the features selected by this method were negatively correlated. For the PCA-based feature selection, support ranged between 0.405 and 0.425 with a total of 38 rules for the first dataset and 36 rules for the second dataset (Table 4.5) In the validation dataset, the maximum support that could generate rules was 0.5 and this means that the rules were positively correlated. Association rules are represented as  $X \Rightarrow Y$ , where X and Y are items contained within a dataset/database, and  $X \cap Y = \emptyset$ . X is the antecedent, and Y is the consequent. It means that whenever X, which is the antecedent, is present, even Y, which is the consequent, will be present. Support indicates the frequency of the itemset appearance in the dataset, and the confidence indicates how often a rule has been found to be true. A support value of 0.5 means 50% of the items (genes) are found in the transaction and 90% of the rules are true (Confidence). Lower support values mean that most of the items are not frequently found together. Lift value is used to measure the rule importance. A lift of greater than 2 achieved by the graph-based feature-selection approach combined with MCC filtering approach indicates the degree to which any two occurrences depend on each other, and this is an indication that those rules are useful in consequent prediction. In association rule mining,

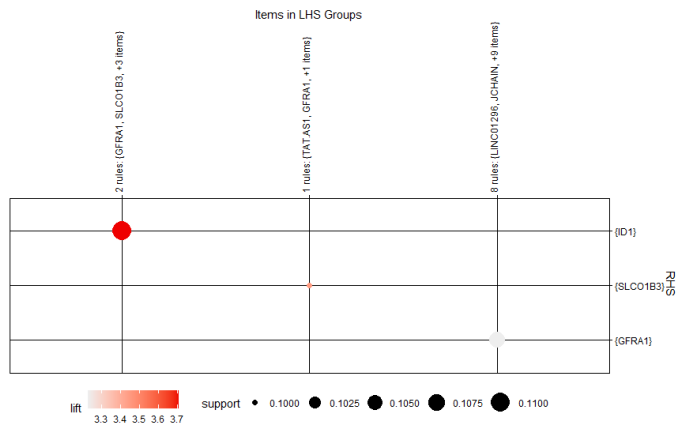
the choice of parameters which are support, confidence and lift really affects the quality and number of rules generated. When the thresholds settings are too high, the rules obtained become very few or none and when the threshold setting is too low, very many redundant rules are generated. Infrequent item sets were utilized by Mahmood *et al.*, (2014) to determine positive and negative association rules. Positive association rule mining removes often occurring things or item sets, but it is possible that many critical items or item sets with low support be rejected. Despite their modest support, these occasional items or item sets can induce significant negative association rules. Although negative association rule mining is crucial, the search space for negative association rule mining is larger than the search space for positive association rule mining since objects with low support must be kept. This makes it substantially difficult for typical Apriori algorithm sequential implementations. In Bagui & Dhar, (2019), experiments were repeated at different support values while keeping the confidence constant at 95% to estimate the appropriate support and confidence values to employ for the trials. The original 1.5 GB dataset, 1 master node, and 5 slave nodes, as well as the default block size of 64 MBs, were employed for this series of trials. Authors tested their method at 15 percent, 20 percent, 30 percent, and 40 percent minimum support levels. Authors reported that for lesser degrees of support and confidence, there are more rules.

Classification was used as an extra validation approach. NB gave the lowest classification accuracy when all features were used for classification. A study by Furat & Ibrici (2019) used five tumor types of gene expression cancer RNA-Seq data and, using Naïve Bayes with 10-fold cross validation, achieved an accuracy of 98.7516%. This shows that NB accuracy levels will vary with the dataset being analyzed. In dataset GSE81089, which had a larger sample size of selected features, SMO and MLP achieved 100% accuracy when feature selection was performed prior to classification, and in fact, the PCA-selected features could be classified at 100% accuracy by all three classifiers. In the smaller dataset, SCLC dataset, accuracy levels were also lower. Notable is that a graph-based feature-selection approach gave the best classification results in the two datasets and took the least time to execute As described in the methodology, four graph filtering techniques which

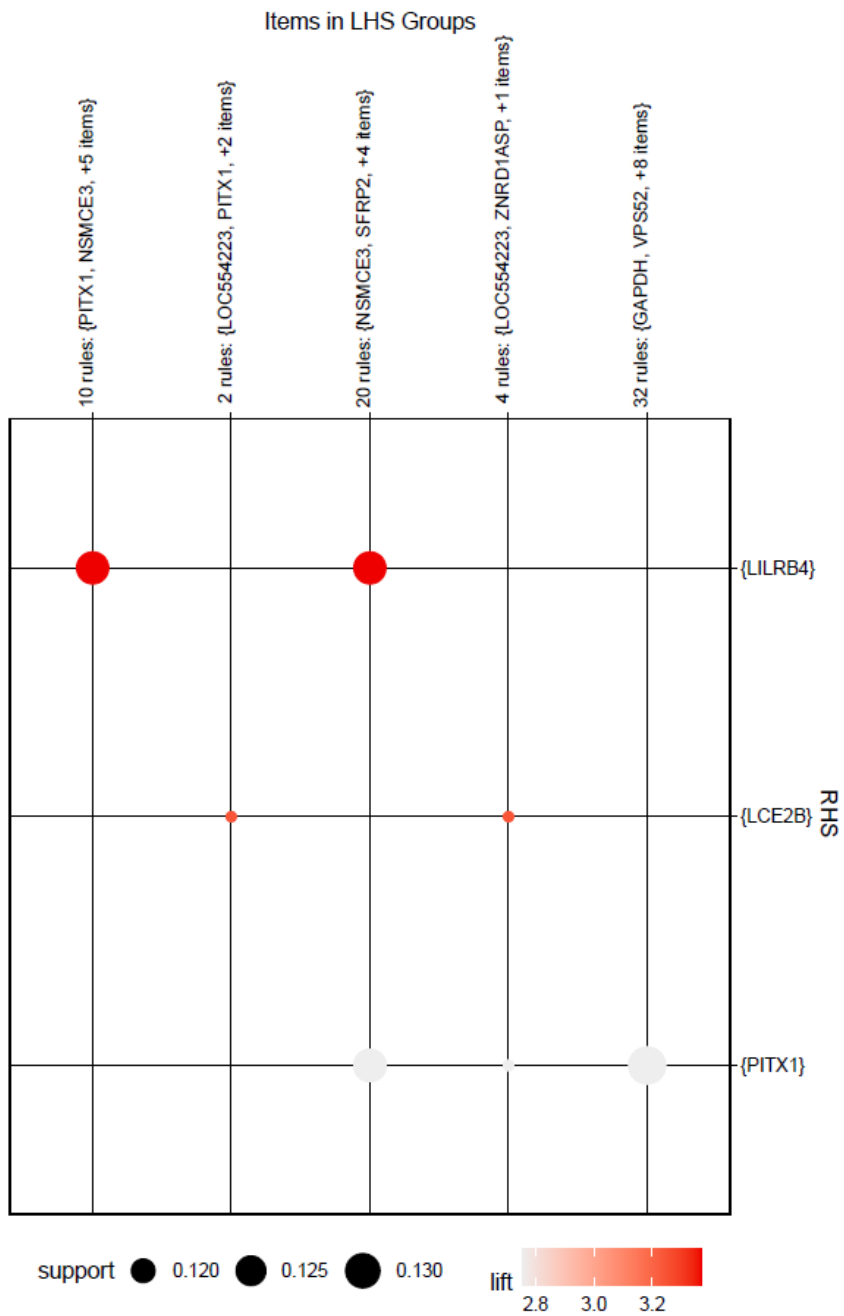
are MCC, EPC, ECC and degree were evaluated. The Graph was filtered using those four approaches and generated rules to test the best approach for graph filtering based on the maximum support of the rules.



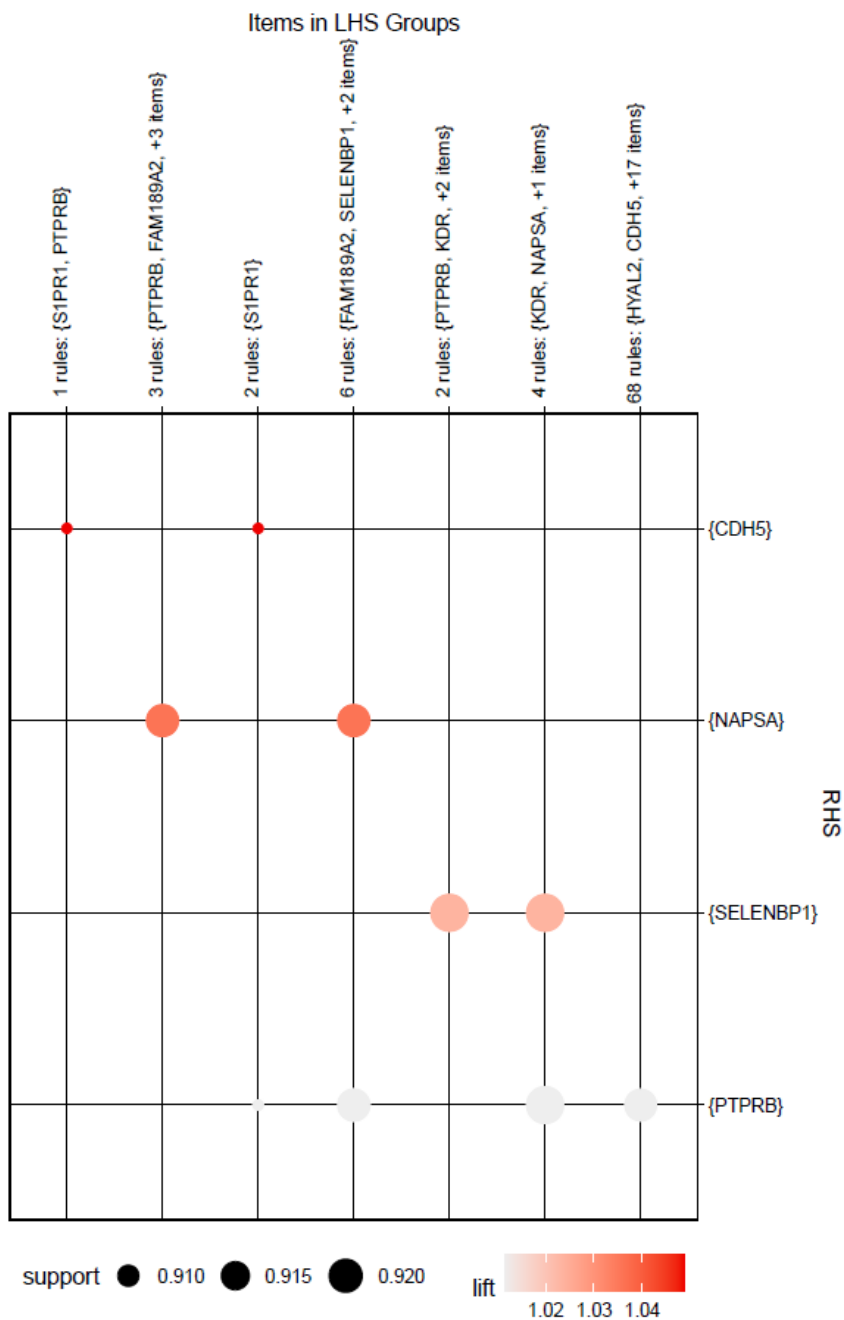
**Figure 4.9a: Summary of rules with lift and support MCC**



**Figure 4.9b: Rules maximum support and lift after filtering using Edge Percolated Component**



**Figure 4.9c: Rules maximum support after filtering using ECC**



**Figure 4.9d: Rules maximum support after filtering using Degree**

ARM results with support and lift for graph filtering techniques represents balloon plot with antecedent grouped as columns and consequents grouped as rows. The total number of antecedents and the most important (frequent) items in the group are displayed as the



labels for the columns. The size of each balloon shows the support value and the bigger the balloon size the larger the support value. The red color of balloons represents the lift and the brighter the color of balloons are, the larger value for the lift in that group. The maximum support of the rules generated after filtering the network using maximal clique was 0.5 and a lift of 2. These rules were positively correlated (Figure 4.9a). However, the maximum support and lift after filtering using Edge Percolated Component and ECC were 0.11 and 0.13 depicting negatively correlated rules (Figure 4.9b, 4.9c). Filtering using degree gave the highest support but the lift was lower than others. This makes maximal clique filtering approach the optimal solution due to the positive correlation of the rules and the high dependency of the frequent item sets as depicted by the lift.

**Table 4.5: A summary of top ten rules generated from the two datasets after graph-based feature selection.**

<b>GSE60052</b>				
<b>Rules</b>	<b>Support</b>	<b>Confidence</b>	<b>Lift</b>	
<b>X</b>	<b>Y</b>			
{SFTPA1, SDC4, LRRK2}	=> {SLC34A2}	0.5	0.9	2
{ACVRL1, COL4A3, AQP1}	=> {SLC34A2}	0.5	0.9	2
{EDNRB, SFTPC, AGER}	=> {SLC34A2}	0.5	0.9	2
{PTPRB, SFTPC, CLDN5}	=> {SLC34A2}	0.5	0.9	2
{EPAS1, EDNRB, LRRK2, AQP1}	=> {SLC34A2}	0.5	0.9	2
{CLDN18, EPAS1, SFTPA1, AGER}	=> {SLC34A2}	0.5	0.9	2
{EPAS1, NAPSA, LRRK2, AGER}	=> {SLC34A2}	0.5	0.9	2
{TIMP3, CTSH, SFTPA1, LRRK2}	=> {SLC34A2}	0.5	0.9	2
{CTSH, NAPSA, TGFBR2, SFTPC}	=> {SLC34A2}	0.5	0.9	2
{RRAS, PTPRB, YAP1, SMAD6}	=> {SLC34A2}	0.5	0.9	2
<b>GSE81089</b>				
<b>Rules</b>	<b>Support</b>	<b>Confidence</b>	<b>Lift</b>	
<b>X</b>	<b>Y</b>			
{ASPM, KIF4A, NUF2}	=> {CENPF}	0.5	1	2
{ASPM, KIF4A, CDC6, NUF2}	=> {TOP2A}	0.5	1	2
{ASPM, CDC6, CDC20, NUF2}	=> {TOP2A}	0.5	1	2
{ASPM, CDC6, CDCA8, NUF2}	=> {TOP2A}	0.5	1	2
{TPX2, FOXM1, NUF2, IQGAP3}	=> {BIRC5}	0.5	1	2
{CDC6, FOXM1, CDC20, UBE2C}	=> {TPX2}	0.5	1	2
{ASPM, CDC6, DLGAP5, NUF2}	=> {TOP2A}	0.5	1	2
{TPX2, CDCA8, UBE2C, IQGAP3}	=> {BIRC5}	0.5	1	2
{ASPM, KIF4A, CDC6, UBE2C}	=> {CENPF}	0.5	1	2
{TPX2, CDC6, FOXM1, IQGAP3}	=> {BIRC5}	0.5	1	2

In transactional databases, relational databases, and other information repositories, the association rule is used to find common patterns, associations, and correlations among collections of items. Frequent item creation and association rule generating processes are crucial in association rule mining. All frequent sets of items are discovered via frequent item generation, which is defined as itemset with at least minimal support. Association rules are generated from these frequently occurring items. The output of graph filtering using MCC and the features generated were used for mining association rules.

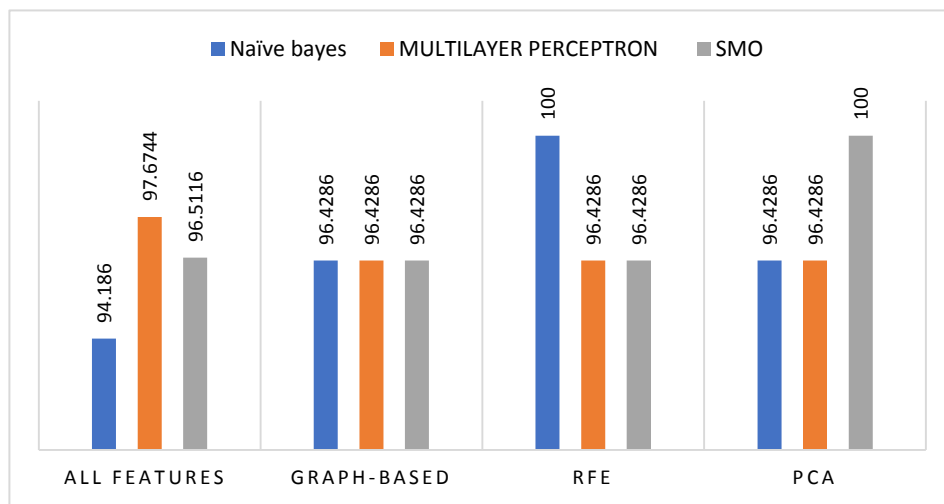
#### **4.7 Classification as an alternative feature selection method**

In the next step, the performance of three classifiers was compared with the features selected in the previous step as input while the raw features were the baseline. Table 4.6 summarizes the performance of the various classifiers before and after feature selection. The accuracy value, root mean squared error (RMSE), mean absolute error (MAE), kappa statistic (KS) and the time taken to build the model for every classifier, arising from the 10-fold cross validation are given. Overall, accuracy levels after selection ranged between 94.186 and 100% depending on the classification method used and the dataset (Table 4.6).

**Table 4.6: Classification results after feature selection.**

<b>NAÏVE BAYES</b>							
<b>Dataset</b>	<b>Feature Selection Method</b>	<b>Accuracy</b>	<b>MAE</b>	<b>Kappa</b>	<b>RMSE</b>	<b>F-Measure</b>	<b>T/s</b>
<b>GSE60052</b>	Graph-based	96.4286	0.0357	0.8679	0.189	0.963	0.01
	RFE	100	0	1	0	1	0.01
	PCA	96.4286	0.0357	0.8679	0.189	0.963	0.02
<b>GSE81089</b>	Graph-based	100	0	1	0	1	0.06
	RFE	100	0	1	0	1	0.01
	PCA	100	0	1	0	1	0.02
<b>MULTILAYER PERCEPTRON</b>							
<b>Dataset</b>	<b>Feature Selection Method</b>	<b>Accuracy</b>	<b>MAE</b>	<b>Kappa</b>	<b>RMSE</b>	<b>F-measure</b>	<b>T/s</b>
<b>GSE60052</b>	Graph-based	96.4286	0.0366	0.8679	0.1814	0.979	18.66
	RFE	96.4286	0.0389	0.8679	0.1851	0.963	9.7
	PCA	96.4286	0.0224	0.8679	0.0993	0.963	9.62
<b>GSE81089</b>	Graph-based	96.4286	0.0389	0.8679	0.1851	0.963	124.15
	RFE	100	0	1	0	1	131.53
	PCA	100	0	1	0	1	0.77
<b>SEQUENTIAL MINIMAL OPTIMIZATION</b>							
<b>Dataset</b>	<b>Feature Selection Method</b>	<b>Accuracy</b>	<b>MAE</b>	<b>Kappa</b>	<b>RMSE</b>	<b>F-measure</b>	<b>T/s</b>
<b>GSE60052</b>	Graph-based	96.4286	0.0357	0.8679	0.189	0.889	0.01
	RFE	96.4286	0.0357	0.8679	0.189	0.963	0.01
	PCA	100	0	1	0	1	0.02
<b>GSE81089</b>	Graph-based	98.5915	0.0141	0.9567	0.1187	0.986	0.14
	RFE	100	0	1	0	1	0.01
	PCA	100	0	1	0	1	0.01

NB performed better on features selected using PCA and a graph-based approach whereby accuracy, MAE, kappa and time taken improved as compared to unfiltered features and RFE-selected features where there was no difference. This can be attributed to the working principle of RFE where the optimal number of features is not known *a priori* (in advance) (Artur,2021). A Kruskal–Wallis test showed that there is no significant difference between the mean ranks of the groups ( $p < 0.05$ ), i.e., 20 iterations for each of the feature-selection methods.



**Figure 4.10: comparison of classifiers accuracy before and after feature selection for dataset GSE60052**

Figure 4.10 shows classifier performance for data set GSE60052 and according to the results, PCA-SMO and RFE-NB gave an accuracy of 100% and to note is the consistency in classifier accuracy for the proposed graph-based feature selection approach across all three classifiers. The time required to build the model improved after feature selection across the three classifiers though MLP required the longest duration and a graph-based approach the shortest (Table 4.6).

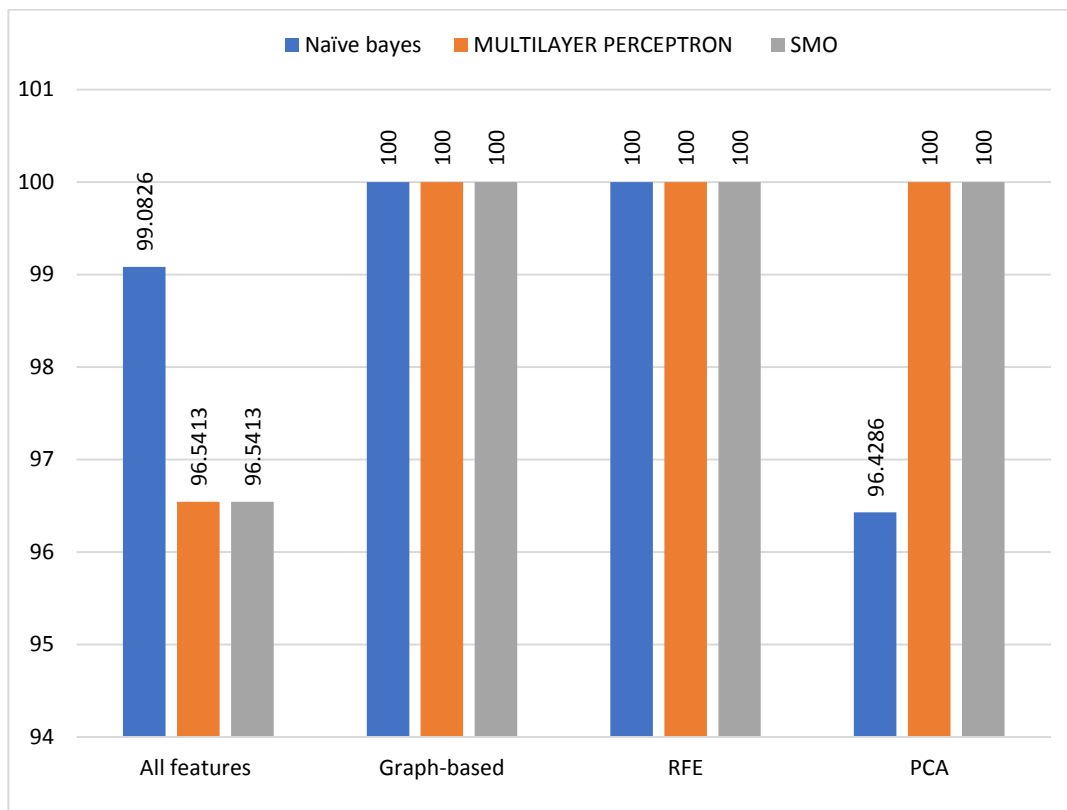


Figure 4.11: **comparison of classifiers accuracy before and after feature selection for dataset GSE81089**

Results for dataset GSE81089 demonstrate that there is clear evidence of model accuracy improvement when graph - based, RFE and PCA based feature selection approaches were used. The findings also revealed a considerable performance difference across feature selection algorithms with accuracy percentages ranging between 96.4 to 100%. As the number of features rises, the classifier performance is affected in terms of time taken to generate a model (Table 4.6). Adlakha & Chhikara, (2016) in their study, increased the data size and reported that the accuracy level of SMO classifier remained the same when there was no feature selection technique is used. Their findings differ from those of this study in that there was an increase of 3.5% in accuracy for SMO and MLP after applying graph based, PCA and RFE feature selection. NB classifier had the highest accuracy when all features were used for classification. Zaffar *et al.*, (2017) used student dataset and reported no significant change in classifiers when feature selection algorithms were

applied other than the principal components feature selection approach when it was combined with Random Forest classifier. Mohammed *et al.*, (2020) used three classifiers' algorithms among them NB, and SMO on two different datasets of the breast cancer. Just like the findings in this study, they reported classifier improvement after feature selection before classification. Another improvement in classifiers accuracy after feature selection was reported in Basker *et al.*, (2021) where WBC dataset was used for analysis. SMO classifier was reported to improve from 71.68 % before feature selection to 99.56% after feature selection. Another classifier that was reported to improve in accuracy after feature selection was NB from 86.52% to 99.12 % after feature selection.

## CHAPTER FIVE

### CONCLUSIONS AND RECOMMENDATION FOR FUTURE RESEARCH

#### 5.1 Conclusions

This study proposed a graph-based feature selection model and association rule mining for phenotype prediction in high-dimensional protein data. RNAseq data generated by next generation sequencing technologies is characterized by large volume and velocity. Therefore, the first objective was to analyze techniques for feature extraction and selection in high dimensional RNAseq data. This data is curated for quality and mapped to a reference genome (animal or plant). The mapping step is usually very critical and is determined by the data quality and the bioinformatic data structure used. The window-based algorithm performed better than *k-mer*-based and running sum algorithm at the trimming step while Burrow's Wheeler algorithm accurately mapped the highest number of reads.

For the second objective PCA, RFE and graph-based feature selection methods were evaluated for their performance when selecting features from RNA-Seq data. RNAseq data is derived from living systems and usually the features interact or influence the observed phenotype. Based on the results, it can be concluded that a graph-based feature selection approach was the most suitable method because because 1) only informative features are selected from the high dimensional data based on their associations in the graph (nodes and edges).

The concept of association rule mining derived from market basket analysis is an important data mining approach to extract inherent relationships between itemsets. The features selected using the graph showed strong association based on the rules generated. The findings from this study demonstrate that a graph-based feature selection approach combined with association rule mining can be very useful in biomarker discovery or

disease phenotype prediction based on gene expression levels. This was validated using an independent Cancer dataset.

### **5.3 Recommendations for future work**

In this study RNAseq data was used for analysis. However, we would recommend:

1. Further research using other forms of RNA data such as Single-cell RNA sequencing (scRNA-seq) and time series data. This will cover other types of biological data for the purpose of the proposed model validation.
2. Use of more feature selection and machine learning approaches on RNAseq data and test the correlation of the generated rules. There are many features selection approaches that can be used in dimensionality reduction of the biological data and compare the strength of the rules generated based on the support and confidence measures
3. In vitro validation of the selected features with no assigned biological function by the life scientists to confirm their function. Since this experiment was 'in silico' which is the application of computational approaches to model, predict, and explain biological function. Validation in a wet lab is recommended



## REFERENCES

- Ab Hamid, T. M. T., Sallehuddin, R., Yunus, Z. M., & Ali, A. (2021). Ensemble Based Filter Feature Selection with Harmonize Particle Swarm Optimization and Support Vector Machine for Optimal Cancer Classification. *Machine Learning with Applications*, 5, 100054.
- Abdulrazzaq, M. B., & Saeed, J. N. (2019, April). A comparison of three classification algorithms for handwritten digit recognition. In *2019 International Conference on Advanced Science and Engineering (ICOASE)* (pp. 58-63). IEEE.
- Abouelhoda, M. I., Kurtz, S., & Ohlebusch, E. (2004). Replacing suffix trees with enhanced suffix arrays. *Journal of discrete algorithms*, 2(1), 53-86.
- Adlakha, A., & Chhikara, R. R. (2016, April). Comparative analysis of filter feature selection techniques with different classifiers for image steganalysis. In *2016 International Conference on Computing, Communication and Automation (ICCCA)* (pp. 1122-1127).
- Agapito, G., Guzzi, P. H., & Cannataro, M. (2021). Parallel and distributed association rule mining in life science: A novel parallel algorithm to mine genomics data. *Information Sciences*, 575, 747-761.
- Agrawal, M., Zitnik, M., & Leskovec, J. (2018). Large-scale analysis of disease pathways in the human interactome. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2018: Proceedings of the Pacific Symposium* (pp. 111-122).
- Agrawal, R., & Srikant, R. (1994, September). Fast algorithms for mining association rules. In Proc. 20th int. conf. very large data bases, VLDB (Vol. 1215, pp. 487-499).

- Agrawal, R., Imieliński, T., & Swami, A. (1993, June). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data* (pp. 207-216).
- Les Laboratoires Servier. Servier Medical Art, CC 3.0 <https://creativecommons.org/licenses/by/3.0/>, <https://smart.servier.com/>, 2018
- Ahmadon, M. A. B., & Yamaguchi, S. (2018, October). User Workflow Preference Analysis Based on Confidence and Lift Value of Association Rule. In 2018 IEEE 7th Global Conference on Consumer Electronics (GCCE) (pp. 610-613). IEEE.
- Ai, D., Pan, H., Li, X., Gao, Y., & He, D. (2018). Association rule mining algorithms on high-dimensional datasets. *Artificial Life and Robotics*, 23(3), 420-427.
- Alagukumar, S., & Lawrance, R. (2015). A selective analysis of microarray data using association rule mining. *Procedia Computer Science*, 47, 3-12.
- Alagukumar, S., & Lawrance, R. (2016, January). Classification of microarray gene expression data using associative classification. In *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)* (pp. 1-8). IEEE.
- Alam, T. M. (2019). Identification of Malignant Mesothelioma Risk Factors through Association Rule Mining.
- Alhunaim, S., & Al-Baity, H. H. (2019). On the scalability of machine-learning algorithms for breast cancer prediction in big data context. *IEEE Access*, 7, 91535-91546.
- Alhenawi, E. A., Al-Sayyed, R., Hudaib, A., & Mirjalili, S. (2022). Feature selection methods on gene expression microarray data for cancer classification: A systematic review. *Computers in biology and medicine*, 140, 105051.

- Ali, H., Salleh, M. N. M., Saedudin, R., Hussain, K., & Mushtaq, M. F. (2019). Imbalance class problems in data mining: A review. *Indonesian Journal of Electrical Engineering and Computer Science*, *14*(3), 1560-1571.
- Ali, M. H., & Baiee, W. R. (2021, June). Choosing an Appropriate Feature Selection Method to Enhance Feed-Forward ANN. In *2021 International Conference on Communication & Information Technology (ICICT)* (pp. 86-91). IEEE.
- Alirezanejad, M., Enayatifar, R., Motameni, H., & Nematzadeh, H. (2020). Heuristic filter feature selection methods for medical datasets. *Genomics*, *112*(2), 1173-1181.
- Almasi, M., & Abadeh, M. S. (2015). Rare-PEARs: A new multi objective evolutionary algorithm to mine rare and non-redundant quantitative association rules. *Knowledge-Based Systems*, *89*, 366-384.
- Alsahaf, A., Petkov, N., Shenoy, V., & Azzopardi, G. (2022). A framework for feature selection through boosting. *Expert Systems with Applications*, *187*, 115895.
- AlSumairi, S. B., & Ismail, M. M. B. (2020). X-ray image based pneumonia classification using convolutional neural networks. *ACCENTS Transactions on Image Processing and Computer Vision*, *6*(20), 54.
- Altaf, W., Shahbaz, M., & Guergachi, A. (2017). Applications of association rule mining in health informatics: a survey. *Artificial Intelligence Review*, *47*(3), 313-340.
- Alves, R., Rodriguez-Baena, D. S., & Aguilar-Ruiz, J. S. (2010). Gene association analysis: a survey of frequent pattern mining from gene expression data. *Briefings in Bioinformatics*, *11*(2), 210-224.
- Amala, A., & Emerson, I. A. (2019). Identification of target genes in cancer diseases using protein-protein interaction networks. *Network Modeling Analysis in Health Informatics and Bioinformatics*, *8*(1), 1-13.

- Angadi, S., & Reddy, V. S. (2021). Multimodal sentiment analysis using reliefF feature selection and random forest classifier. *International Journal of Computers and Applications*, 43(9), 931-939.
- Anguita-Ruiz, A., Segura-Delgado, A., Alcalá, R., Aguilera, C. M., & Alcalá-Fdez, J. (2020). eXplainable Artificial Intelligence (XAI) for the identification of biologically relevant gene expression patterns in longitudinal human studies, insights from obesity research. *PLoS computational biology*, 16(4), e1007792.
- Arabnejad, M., Dawkins, B. A., Bush, W. S., White, B. C., Harkness, A. R., & McKinney, B. A. (2018). Transition-transversion encoding and genetic relationship metric in ReliefF feature selection improves pathway enrichment in GWAS. *BioData mining*, 11(1), 1-17.
- Arora, S. (2019). Data Science vs. Big Data vs. Data Analytics. *Elérhető: www.simplilearn.com/data-science-vs-big-data-vs-data-analytics-article (A letöltés dátuma: 2019. 03. 29.)*.
- Arowolo, M. O., Adebisi, M. O., Adebisi, A. A., & Olugbara, O. (2021). Optimized hybrid investigative based dimensionality reduction methods for malaria vector using KNN classifier. *Journal of Big Data*, 8(1), 1-14.
- Arowolo, M. O., Isiaka, R. M., Abdulsalam, S. O., Saheed, Y. K., & Gbolagade, K. A. (2017). A comparative analysis of feature extraction methods for classifying colon cancer microarray data. *EAI endorsed transactions on scalable information systems*, 4(14).
- Artur, M. (2021). Review the performance of the Bernoulli Naïve Bayes Classifier in Intrusion Detection Systems using Recursive Feature Elimination with Cross-validated selection of the best number of features. *Procedia Computer Science*, 190, 564-570.

- Ashiqur Rahman, S., Giacobbi, P., Pyles, L., Mullett, C., Doretto, G., & Adjeroh, D. A. (2021). Deep learning for biological age estimation. *Briefings in bioinformatics*, 22(2), 1767-1781.
- Aziz, R., Verma, C. K., & Srivastava, N. (2018). Artificial neural network classification of high dimensional data with novel optimization approach of dimension reduction. *Annals of Data Science*, 5(4), 615-635.
- Bader-El-Den, M., Teitei, E., & Perry, T. (2018). Biased random forest for dealing with the class imbalance problem. *IEEE transactions on neural networks and learning systems*, 30(7), 2163-2172.
- Bagui, S., & Dhar, P. C. (2019). Positive and negative association rule mining in Hadoop's MapReduce environment. *Journal of Big Data*, 6(1), 1-16.
- Bai, A., & Hira, S. (2021). Microarray cancer classification using feature extraction-based ensemble learning method. *International Journal of Data Analysis Techniques and Strategies*, 13(3), 244-263.
- Banka, H., & Dara, S. (2015). A Hamming distance based binary particle swarm optimization (HDBPSO) algorithm for high dimensional feature selection, classification and validation. *Pattern Recognition Letters*, 52, 94-100.
- Bansal, A., Srivastava, P. A., & Singh, T. R. (2018). An integrative approach to develop computational pipeline for drug-target interaction network analysis. *Scientific reports*, 8(1), 1-9.
- Banuchitra, S. (2021). Image Retrieval Using Hierarchical Nested Clusters. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(13), 2499-2510.

- Basker, N., Theetchenya, S., Vidyabharathi, D., Dhaynithi, J., Mohanraj, G., Marimuthu, M., & Vidhya, G. (2021). Breast Cancer Detection Using Machine Learning Algorithms. *Annals of the Romanian Society for Cell Biology*, 2551-2562.
- Behzadi, P., & Ranjbar, R. (2019). DNA microarray technology and bioinformatic web services. *Acta microbiologica et immunologica Hungarica*, 66(1), 19-30.
- Beiranvand, V., Mobasher-Kashani, M., & Bakar, A. A. (2014). Multi-objective PSO algorithm for mining numerical association rules without a priori discretization. *Expert systems with applications*, 41(9), 4259-4273.
- Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1), 289-300.
- Bhattacharya, S., Maddikunta, P. K. R., Kaluri, R., Singh, S., Gadekallu, T. R., Alazab, M., & Tariq, U. (2020). A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU. *Electronics*, 9(2), 219.
- Bhavsar, H., & Ganatra, A. (2012). A comparative study of training algorithms for supervised machine learning. *International Journal of Soft Computing and Engineering (IJSCE)*, 2(4), 2231-2307.
- Bi, X., Zhao, X., Huang, H., Chen, D., & Ma, Y. (2020). Functional brain network classification for Alzheimer's disease detection with deep features and extreme learning machine. *Cognitive Computation*, 12(3), 513-527.
- Bobrov, E., Georgievskaya, A., Kiselev, K., Sevastopolsky, A., Zhavoronkov, A., Gurov, S., ... & Clemann, S. (2018). PhotoAgeClock: deep learning algorithms for development of non-invasive visual biomarkers of aging. *Aging (Albany NY)*, 10(11), 3249.

- Bolón-Canedo, V. (2014). Novel feature selection methods for high dimensional data.
- Bolón-Canedo, V., Sánchez-Marño, N., & Alonso-Betanzos, A. (2016). Feature selection for high-dimensional data. *Progress in Artificial Intelligence*, 5(2), 65-75.
- Bommert, A., Sun, X., Bischl, B., Rahnenführer, J., & Lang, M. (2020). Benchmark for filter methods for feature selection in high-dimensional classification data. *Computational Statistics & Data Analysis*, 143, 106839.
- Bossaghzadeh, A. (2020, February). Improving persian digit recognition by combining deep neural networks and SVM and using PCA. In *2020 International Conference on Machine Vision and Image Processing (MVIP)* (pp. 1-5). IEEE.
- Bray, N., Pimentel, H., Melsted, P., & Pachter, L. (2016). Near-optimal RNA-Seq quantification with kallisto. *Nat Biotechnol*, 34(5), 525-527.
- Brinda, K. (2016). *Novel computational techniques for mapping and classification of Next-Generation Sequencing data* (Doctoral dissertation, Université Paris-Est).
- Brosnan, J. T., & Brosnan, M. E. (2006). The sulfur-containing amino acids: an overview. *The Journal of nutrition*, 136(6), 1636S-1640S.
- Busch, A., Homeier-Bachmann, T., Abdel-Glil, M. Y., Hackbart, A., Hotzel, H., & Tomaso, H. (2020). Using affinity propagation clustering for identifying bacterial clades and subclades with whole-genome sequences of *Francisella tularensis*. *PLoS neglected tropical diseases*, 14(9), e0008018.
- Bustamam, A., Formalidin, S., & Siswantining, T. (2018, October). Clustering and analyzing microarray data of lymphoma using singular value decomposition (SVD) and hybrid clustering. In *AIP Conference Proceedings* (Vol. 2023, No. 1, p. 020220). AIP Publishing LLC.

- Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE). *Geoscientific Model Development Discussions*, 7(1), 1525-1534.
- Chang, J., & Wang, H. (2018, April). Video Segmentation by Spatio-temporal Random Walk. In *Proceedings of the 2018 International Conference on E-Business, Information Management and Computer Science* (pp. 54-58).
- Chartrand, G., Haynes, T. W., Henning, M. A., & Zhang, P. (2019). Eulerian and Hamiltonian Walks. In *From Domination to Coloring* (pp. 57-68). Springer, Cham.
- Chen, C., Wu, W., Chen, C., Chen, F., Dong, X., Ma, M., ... & Zhu, M. (2021). Rapid diagnosis of lung cancer and glioma based on serum Raman spectroscopy combined with deep learning. *Journal of Raman Spectroscopy*, 52(11), 1798-1809.
- Chen, J. W., & Dhahbi, J. (2021). Lung adenocarcinoma and lung squamous cell carcinoma cancer classification, biomarker identification, and gene expression analysis using overlapping feature selection methods. *Scientific Reports*, 11(1), 1-15.
- Chen, J., Li, Z., & Huang, B. (2017). Linear spectral clustering superpixel. *IEEE Transactions on image processing*, 26(7), 3317-3330.
- Chen, S., Li, Z., Pan, G., & Xu, F. (2022). Power Quality Disturbance Recognition Using Empirical Wavelet Transform and Feature Selection. *Electronics*, 11(2), 174.
- Cheng, F., Desai, R. J., Handy, D. E., Wang, R., Schneeweiss, S., Barabási, A. L., & Loscalzo, J. (2018). Network-based approach to prediction and population-based validation of in silico drug repurposing. *Nature communications*, 9(1), 1-12.



- Chiang, M., & Yang, M. (2004, September). Towards network X-ities from a topological point of view: Evolvability and scalability. In *Proc., Allerton Conf. on Comm., Control, and Computing*.
- Chiclana, F., Kumar, R., Mittal, M., Khari, M., Chatterjee, J. M., & Baik, S. W. (2018). ARM-AMO: an efficient association rule mining algorithm based on animal migration optimization. *Knowledge-Based Systems, 154*, 68-80.
- Chikhi, R., Holub, J., & Medvedev, P. (2021). Data structures to represent a set of k-long DNA sequences. *ACM Computing Surveys (CSUR), 54*(1), 1-22.
- Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., ... & Mortazavi, A. (2016). A survey of best practices for RNA-seq data analysis. *Genome biology, 17*(1), 1-19.
- Cribben, I., & Yu, Y. (2017). Estimating whole- brain dynamics by using spectral clustering. *Journal of the Royal Statistical Society: Series C (Applied Statistics), 66*(3), 607-627.
- Curtin, K., Fleckenstein, A. E., Robison, R. J., Crookston, M. J., Smith, K. R., & Hanson, G. R. (2015). Methamphetamine/amphetamine abuse and risk of Parkinson's disease in Utah: a population-based assessment. *Drug and alcohol dependence, 146*, 30-38.
- Das, K., Samanta, S., & Pal, M. (2018). Study on centrality measures in social networks: a survey. *Social network analysis and mining, 8*(1), 1-11.
- de Holanda Maia, M. R., Plastino, A., & Freitas, A. A. (2021, December). An ensemble of naïve Bayes classifiers for uncertain categorical data. In *2021 IEEE International Conference on Data Mining (ICDM)* (pp. 1222-1227). IEEE.

- Deelen, P., van Dam, S., Herkert, J. C., Karjalainen, J. M., Brugge, H., Abbott, K. M., ... & Franke, L. (2019). Improving the diagnostic yield of exome-sequencing by predicting gene–phenotype associations using large-scale gene expression analysis. *Nature communications*, *10*(1), 1-13.
- Degli Esposti, D., Almunia, C., Guery, M. A., Koenig, N., Armengaud, J., Chaumot, A., & Geffard, O. (2019). Co-expression network analysis identifies gonad-and embryo-associated protein modules in the sentinel species *Gammarus fossarum*. *Scientific reports*, *9*(1), 1-10.
- Del Fabbro, C., Scalabrin, S., Morgante, M., & Giorgi, F. M. (2013). An extensive evaluation of read trimming effects on Illumina NGS data analysis. *PLoS one*, *8*(12), e85024.
- Deng, J., Guo, J., & Wang, Y. (2019). A Novel K-medoids clustering recommendation algorithm based on probability distribution for collaborative filtering. *Knowledge-Based Systems*, *175*, 96-106.
- Desai, M., & Shah, M. (2021). An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN). *Clinical eHealth*, *4*, 1-11.
- Dhahbi, J. M. (2021). LUAD And LUSC Cancer Classification, Biomarker Identification, and Pathway Analysis Using Overlapping Feature Selection Methods.
- Dhalmahapatra, K., Shingade, R., & Maiti, J. (2020). An innovative integrated modelling of safety data using multiple correspondence analysis and fuzzy discretization techniques. *Safety Science*, *130*, 104828.
- Djureinovic, D., Hallström, B. M., Horie, M., Mattsson, J. S. M., La Fleur, L., Fagerberg, L., ... & Micke, P. (2016). Profiling cancer testis antigens in non–small-cell lung cancer. *JCI insight*, *1*(10).

- Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., ... & Gingeras, T. R. (2013). STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1), 15-21.
- Dong, X., & Liu, C. (2015). Mining interesting infrequent and frequent itemsets based on multiple level minimum supports and minimum correlation strength. *International Journal of Services Technology and Management*, 21(4-6), 301-317.
- Douzas, G., Bacao, F., & Last, F. (2018). Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Information Sciences*, 465, 1-20.
- Drouin, A., Letarte, G., Raymond, F., Marchand, M., Corbeil, J., & Laviolette, F. (2019). Interpretable genotype-to-phenotype classifiers with performance guarantees. *Scientific reports*, 9(1), 1-13.
- Dua, S., & Chowriappa, P. (2012). *Data mining for bioinformatics*. CRC Press.
- Dutta, A. K., Elhoseny, M., Dahiya, V., & Shankar, K. (2020). An efficient hierarchical clustering protocol for multihop Internet of vehicles communication. *Transactions on Emerging Telecommunications Technologies*, 31(5), e3690.
- Dutta, P., Basu, S., & Kundu, M. (2017). Assessment of semantic similarity between proteins using information content and topological properties of the gene ontology graph. *IEEE/ACM transactions on computational biology and bioinformatics*, 15(3), 839-849.
- Elhilbawi, H., Eldawlatly, S., & Mahdi, H. (2021, March). The importance of discretization methods in machine learning applications: A case study of predicting ICU mortality. In *International Conference on Advanced Machine Learning Technologies and Applications* (pp. 214-224). Springer, Cham.

- Elsakaan, N., & Amroun, K. (2021, November). A Comparative Study of Machine Learning Binary Classification Methods for Botnet Detection. In *International Conference on Applied CyberSecurity* (pp. 20-34). Springer, Cham.
- Eluri, N. R. (2021). Feature Extraction In Gene Expression Dataset Using Multilayer Perceptron. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(2), 3069-3076.
- Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd* (Vol. 96, No. 34, pp. 226-231).
- Eswari, T., Sampath, P., & Lavanya, S. J. P. C. S. (2015). Predictive methodology for diabetic data analysis in big data. *Procedia Computer Science*, 50, 203-208.
- Ewing, B., & Green, P. (1998). Base-calling of automated sequencer traces using phred. II. Error probabilities. *Genome research*, 8(3), 186-194.
- Fahrudin, T. M., Syarif, I., & Barakbah, A. R. (2016, October). Feature selection algorithm using information gain based clustering for supporting the treatment process of breast cancer. In *2016 International Conference on Informatics and Computing (ICIC)* (pp. 6-11). IEEE.
- Feng, J., Chen, J., Sun, Q., Shang, R., Cao, X., Zhang, X., & Jiao, L. (2020). Convolutional neural network based on bandwise-independent convolution and hard thresholding for hyperspectral band selection. *IEEE Transactions on Cybernetics*, 51(9), 4414-4428.
- Feng, S., Heath, E., Jefferson, B., Joslyn, C., Kvinge, H., Mitchell, H. D., ... & Purvine, E. (2021). Hypergraph models of biological networks to identify genes critical to pathogenic viral response. *BMC bioinformatics*, 22(1), 1-21.

- Feng, Y., Lu, B., & Zhang, D. (2017). Multiscale morphological manifold for rolling bearing fault diagnosis. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 231(19), 3516-3529.
- Fikri, M. N., Hassan, M. F., & Tran, D. C. (2020). The impact of fuzzy discretization's output on classification accuracy of random forest classifier. *International Journal*, 9(3).
- Fonseca, L., Pul, C. V., Lori, N., Boom, R., Andriessen, P., Buijs, J., & Vilanova, A. (2017). Automatic atlas-based segmentation of brain white matter in neonates at risk for neurodevelopmental disorders. In *Modeling, analysis, and visualization of anisotropy* (pp. 355-372). Springer, Cham.
- Fu, G. H., Wu, Y. J., Zong, M. J., & Pan, J. (2020). Hellinger distance-based stable sparse feature selection for high-dimensional class-imbalanced data. *BMC bioinformatics*, 21(1), 1-14.
- Fujisawa, K., Shimo, M., Taguchi, Y. H., Ikematsu, S., & Miyata, R. (2021). PCA-based unsupervised feature extraction for gene expression analysis of COVID-19 patients. *Scientific reports*, 11(1), 1-11.
- Furat, F. G., & İbrikçi, T. (2019). Tumor Type Detection Using Naïve Bayes Algorithm on Gene Expression Cancer RNA-Seq Data Set. *Lung Cancer*, 10, 13.
- Gadekallu, T. R., Khare, N., Bhattacharya, S., Singh, S., Maddikunta, P. K. R., Ra, I. H., & Alazab, M. (2020). Early detection of diabetic retinopathy using PCA-firefly based deep learning model. *Electronics*, 9(2), 274.
- Gallo, C. A., Cecchini, R. L., Carballido, J. A., Micheletto, S., & Ponzoni, I. (2016). Discretization of gene expression data revised. *Briefings in bioinformatics*, 17(5), 758-770.

- Ganegoda, G. U., Sheng, Y., & Wang, J. (2015). ProSim: a method for prioritizing disease genes based on protein proximity and disease similarity. *Biomed Res Int*, 2015(5), 213750.
- Gao, L., Sun, P. G., & Song, J. (2009). Clustering algorithms for detecting functional modules in protein interaction networks. *Journal of Bioinformatics and Computational Biology*, 7(01), 217-242.
- García, S., Luengo, J., & Herrera, F. (2015). *Data preprocessing in data mining* (Vol. 72, pp. 59-139). Cham, Switzerland: Springer International Publishing.
- Gärtner, B., & Hiebl, M. R. (2017). Issues with big data. In *The Routledge companion to accounting information systems* (pp. 161-172). Routledge.
- Ghaemi, M., & Feizi-Derakhshi, M. R. (2016). Feature selection using forest optimization algorithm. *Pattern Recognition*, 60, 121-129.
- Ghojogh, B., Ghodsi, A., Karray, F., & Crowley, M. (2021). Factor analysis, probabilistic principal component analysis, variational inference, and variational autoencoder: Tutorial and survey. *arXiv preprint arXiv:2101.00734*..
- Gobin, E., Bagwell, K., Wagner, J., Mysona, D., Sandirasegarane, S., Smith, N., ... & She, J. X. (2019). A pan-cancer perspective of matrix metalloproteases (MMP) gene expression profile and their diagnostic/prognostic potential. *BMC cancer*, 19(1), 1-10.
- Gog, S., Beller, T., Moffat, A., & Petri, M. (2014, June). From theory to practice: Plug and play with succinct data structures. In *International Symposium on Experimental Algorithms* (pp. 326-337). Springer, Cham.

- Govender, P., & Sivakumar, V. (2020). Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019). *Atmospheric Pollution Research*, 11(1), 40-56.
- Granik, M., & Mesyura, V. (2017, May). Fake news detection using Naïve Bayes classifier. In *2017 IEEE first Ukraine conference on electrical and computer engineering (UKRCON)* (pp. 900-903). IEEE.
- Grolinger, K., Hayes, M., Higashino, W. A., L'Heureux, A., Allison, D. S., & Capretz, M. A. (2014, June). Challenges for mapreduce in big data. In *2014 IEEE world congress on services* (pp. 182-189). IEEE.
- Gross, J. L., Yellen, J., & Anderson, M. (2018). *Graph theory and its applications*. Chapman and Hall/CRC.
- Grunspan, D. Z., Wiggins, B. L., & Goodreau, S. M. (2014). Understanding classrooms through social network analysis: A primer for social network analysis in education research. *CBE—Life Sciences Education*, 13(2), 167-178.
- Gunaratne, C., Reyes, R., Hemberg, E., & O'Reilly, U. M. (2021). Evaluating Efficacy of Indoor Non-Pharmaceutical Interventions against COVID-19 Outbreaks with a Coupled Spatial-SIR Agent-Based Simulation Framework. *arXiv preprint arXiv:2108.11025*.
- Guo, Y., Chung, F. L., Li, G., & Zhang, L. (2019). Multi-label bioinformatics data classification with ensemble embedded feature selection. *IEEE Access*, 7, 103863-103875.
- Hacibeyoğlu, m., & ibrahim, M. H. (2016). Comparison of the effect of unsupervised and supervised discretization methods on classification process. *International Journal of Intelligent Systems and Applications in Engineering*, 105-108.

- Halder, A. K., Denkwicz, M., Sengupta, K., Basu, S., & Plewczynski, D. (2020). Aggregated network centrality shows non-random structure of genomic and proteomic networks. *Methods*, *181*, 5-14.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, *11*(1), 10-18.
- Hameed, S. S., Petinrin, O. O., Hashi, A. O., & Saeed, F. (2018). Filter-wrapper combination and embedded feature selection for gene expression data. *Int. J. Advance Soft Compu. Appl*, *10*(1), 90-105.
- Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. *ACM sigmod record*, *29*(2), 1-12.
- Hasan, B. M. S., & Abdulazeez, A. M. (2021). A review of principal component analysis algorithm for dimensionality reduction. *Journal of Soft Computing and Data Mining*, *2*(1), 20-30.
- Hasanin, T., Khoshgoftaar, T. M., Leevy, J., & Seliya, N. (2019, April). Investigating random undersampling and feature selection on bioinformatics big data. In *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)* (pp. 346-356). IEEE.
- Hashemi, A., Dowlatshahi, M. B., & Nezamabadi-Pour, H. (2020). MGFS: A multi-label graph-based feature selection algorithm via PageRank centrality. *Expert Systems with Applications*, *142*, 113024.
- Havaei, M., Guizard, N., Larochelle, H., & Jodoin, P. M. (2016). Deep learning trends for focal brain pathology segmentation in MRI. In *Machine learning for health informatics* (pp. 125-148). Springer, Cham.



- He, C., Du, Y., Fu, J., Zeng, E., Park, S., White, F., ... & Liu, S. (2020). Early drought-responsive genes are variable and relevant to drought tolerance. *G3: Genes, Genomes, Genetics*, *10*(5), 1657-1670.
- He, Y., Shen, Z., Zhang, Q., Wang, S., & Huang, D. S. (2021). A survey on deep learning in DNA/RNA motif mining. *Briefings in Bioinformatics*, *22*(4), bbaa229.
- Henkel, L., Rauscher, B., & Boutros, M. (2019). Context-dependent genetic interactions in cancer. *Current Opinion in Genetics & Development*, *54*, 73-82.
- Henni, K., Mezghani, N., & Gouin-Vallerand, C. (2018). Unsupervised graph-based feature selection via subspace and pagerank centrality. *Expert Systems with Applications*, *114*, 46-53.
- Hira, Z. M., & Gillies, D. F. (2015). A review of feature selection and feature extraction methods applied on microarray data. *Advances in bioinformatics*, *2015*.
- Hobbs, B. D., Morrow, J. D., Celli, B. R., Bueno, R., Criner, G. J., DeMeo, D. L., ... & Cho, M. H. (2017). Chronic obstructive pulmonary disease subtyping through multiple-omics data integration. In *C21. Omics in lung disease* (pp. A4964-A4964). American Thoracic Society.
- Hou, J., Ye, X., Feng, W., Zhang, Q., Han, Y., Liu, Y., ... & Wei, Y. (2022). Distance correlation application to gene co-expression network analysis. *BMC bioinformatics*, *23*(1), 1-24.
- Hou, R., Denisenko, E., & Forrest, A. R. (2019). scMatch: a single-cell gene expression profile annotation tool using reference datasets. *Bioinformatics*, *35*(22), 4688-4695.

- Hranisavljevic, N., Maier, A., & Niggemann, O. (2020). Discretization of hybrid CPPS data into timed automaton using restricted Boltzmann machines. *Engineering Applications of Artificial Intelligence*, 95, 103826.
- Hu, D., Nie, F., & Li, X. (2018). Discrete spectral hashing for efficient similarity retrieval. *IEEE Transactions on Image Processing*, 28(3), 1080-1091.
- Hu, L., & Cui, J. (2019). Digital image recognition based on Fractional-order-PCA-SVM coupling algorithm. *Measurement*, 145, 150-159.
- Hu, Q., Si, X. S., Qin, A. S., Lv, Y. R., & Zhang, Q. H. (2020). Machinery fault diagnosis scheme using redefined dimensionless indicators and mRMR feature selection. *IEEE Access*, 8, 40313-40326.
- Hu, R., Zhu, X., Cheng, D., He, W., Yan, Y., Song, J., & Zhang, S. (2017). Graph self-representation method for unsupervised feature selection. *Neurocomputing*, 220, 130-137.
- Huang, T., Lan, L., Fang, X., An, P., Min, J., & Wang, F. (2015). Promises and challenges of big data computing in health sciences. *Big Data Research*, 2(1), 2-11.
- Hughes, G., Kopetzky, J., & McRoberts, N. (2020). Mutual information as a performance measure for binary predictors characterized by both ROC curve and PROC curve analysis. *Entropy*, 22(9), 938.
- Huysmans, D. (2021). Advances in unobtrusive monitoring of sleep apnea using machine learning.
- Hwang, S., Kim, C. Y., Yang, S., Kim, E., Hart, T., Marcotte, E. M., & Lee, I. (2019). HumanNet v2: human gene networks for disease research. *Nucleic acids research*, 47(D1), D573-D580.

- Indhumathy, M., Nabhan, A. R., & Arumugam, S. (2018). A weighted association rule mining method for predicting HCV-human protein interactions. *Current Bioinformatics*, 13(1), 73-84.
- Jamal, A., Handayani, A., Septiandri, A. A., Ripmiatin, E., & Effendi, Y. (2018). Dimensionality reduction using pca and k-means clustering for breast cancer prediction. *Lontar Komput. J. Ilm. Teknol. Inf*, 9(3), 192-201.
- Jardim, V. C., Santos, S. D. S., Fujita, A., & Buckeridge, M. S. (2019). BioNetStat: a tool for biological networks differential analysis. *Frontiers in genetics*, 594.
- Jayaweera, I. M. L. N., Perera, K. K. K. R., & Munasinghe, J. (2017). Centrality measures to identify traffic congestion on road networks: A case study of sri lanka. *IOSR Journal of Mathematics (IOSRJM)*.
- Jiang, L., Huang, J., Higgs, B. W., Hu, Z., Xiao, Z., Yao, X., ... & Yao, Y. (2016). Genomic landscape survey identifies SRSF1 as a key oncodriver in small cell lung cancer. *PLoS genetics*, 12(4), e1005895.
- Jiang, M., Chen, Y., & Chen, L. (2015). Link prediction in networks with nodes attributes by similarity propagation. *arXiv preprint arXiv:1502.04380*.
- Jindal, P., & Kumar, D. (2017). A review on dimensionality reduction techniques. *International journal of computer applications*, 173(2), 42-46.
- Jo, I., Lee, S., & Oh, S. (2019). Improved measures of redundancy and relevance for mRMR feature selection. *Computers*, 8(2), 42.
- Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255-260.

- Kaletsky, R., Yao, V., Williams, A., Runnels, A. M., Tadych, A., Zhou, S., ... & Murphy, C. T. (2018). Transcriptome analysis of adult *Caenorhabditis elegans* cells reveals tissue-specific gene and isoform expression. *PLoS genetics*, *14*(8), e1007559.
- Kanitz, A., Gypas, F., Gruber, A. J., Gruber, A. R., Martin, G., & Zavolan, M. (2015). Comparative assessment of methods for the computational inference of transcript isoform abundance from RNA-seq data. *Genome biology*, *16*(1), 1-26.
- Karim, M. R., Beyan, O., Zappa, A., Costa, I. G., Rebholz-Schuhmann, D., Cochez, M., & Decker, S. (2021). Deep learning-based clustering approaches for bioinformatics. *Briefings in Bioinformatics*, *22*(1), 393-415.
- Kaur, P., & Gosain, A. (2018). Comparing the behavior of oversampling and undersampling approach of class imbalance learning by combining class imbalance problem with noise. In *ICT Based Innovations* (pp. 23-30). Springer, Singapore.
- Kaya, I. E., Pehlivanlı, A. Ç., Sekizkardeş, E. G., & Ibrikci, T. (2017). PCA based clustering for brain tumor segmentation of T1w MRI images. *Computer methods and programs in biomedicine*, *140*, 19-28.
- Khare, K., Oh, S. Y., Rahman, S., & Rajaratnam, B. (2019). A scalable sparse Cholesky based approach for learning high-dimensional covariance matrices in ordered data. *Machine Learning*, *108*(12), 2061-2086.
- Kim, J., Ji, M., & Yi, G. (2020). A review on sequence alignment algorithms for short reads based on next-generation sequencing. *IEEE Access*, *8*, 189811-189822.
- Kim, K. (2018). An improved semi-supervised dimensionality reduction using feature weighting: application to sentiment analysis. *Expert Systems with Applications*, *109*, 49-65.

- Kim, S., Kang, D., Huo, Z., Park, Y., & Tseng, G. C. (2018). Meta-analytic principal component analysis in integrative omics application. *Bioinformatics*, *34*(8), 1321-1328.
- Kloetgen, A., Muench, P. C., Borkhardt, A., Hoell, J. I., & McHardy, A. C. (2015). Biochemical and bioinformatic methods for elucidating the role of RNA–protein interactions in posttranscriptional regulation. *Briefings in functional genomics*, *14*(2), 102-114.
- Köhler, S., Bauer, S., Horn, D., & Robinson, P. N. (2008). Walking the interactome for prioritization of candidate disease genes. *The American Journal of Human Genetics*, *82*(4), 949–958
- Kolde, R. (2012). Pheatmap: pretty heatmaps. *R package version*, *1*(2), 747.
- Kothandan, R. (2015). Handling class imbalance problem in miRNA dataset associated with cancer. *Bioinformation*, *11*(1), 6.
- Kucherov, G., Tóthmérész, L., & Vialette, S. (2013). On the combinatorics of suffix arrays. *Information Processing Letters*, *113*(22-24), 915-920.
- Kumar, K. A., Gluck, J., Deshpande, A., & Lin, J. (2013). Hone: "Scaling down" Hadoop on shared-memory systems. *Proceedings of the VLDB Endowment*, *6*(12), 1354-1357.
- Kumar, V., & Minz, S. (2014). Feature selection: a literature review. *SmartCR*, *4*(3), 211-229.
- L'heureux, A., Grolinger, K., Elyamany, H. F., & Capretz, M. A. (2017). Machine learning with big data: Challenges and approaches. *Ieee Access*, *5*, 7776-7797.

- Lall, S., Sinha, D., Ghosh, A., Sengupta, D., & Bandyopadhyay, S. (2021). Stable feature selection using copula based mutual information. *Pattern Recognition*, *112*, 107697.
- Lauria, A., Peirone, S., Giudice, M. D., Priante, F., Rajan, P., Caselle, M., ... & Cereda, M. (2020). Identification of altered biological processes in heterogeneous RNA-sequencing data by discretization of expression profiles. *Nucleic acids research*, *48*(4), 1730-1747.
- Lazzarini, N., Runhaar, J., Bay-Jensen, A. C., Thudium, C. S., Bierma-Zeinstra, S. M. A., Henrotin, Y., & Bacardit, J. (2017). A machine learning approach for the identification of new biomarkers for knee osteoarthritis development in overweight and obese women. *Osteoarthritis and cartilage*, *25*(12), 2014-2021.
- Le, D. H., & Pham, V. H. (2017). HGPEC: a Cytoscape app for prediction of novel disease-gene and disease-disease associations and evidence collection based on a random walk on heterogeneous network. *BMC systems biology*, *11*(1), 1-10.
- Le, E. P. V., Wang, Y., Huang, Y., Hickman, S., & Gilbert, F. J. (2019). Artificial intelligence in breast imaging. *Clinical radiology*, *74*(5), 357-366.
- Lei, X., & Bian, C. (2020). Integrating random walk with restart and k-Nearest Neighbor to identify novel circRNA-disease association. *Scientific reports*, *10*(1), 1-9.
- Lei, X., Yang, X., & Fujita, H. (2019). Random walk based method to identify essential proteins by integrating network topology and biological characteristics. *Knowledge-Based Systems*, *167*, 53-67.
- Li, H., Zhao, J., Zhang, X., Teng, H., Yang, R., & Hao, L. (2014). Bearing fault diagnosis method using envelope analysis and euclidean distance. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, *12*(3), 1887-1894.

- Li, L., Wang, Y., An, L., Kong, X., & Huang, T. (2017). A network-based method using a random walk with restart algorithm and screening tests to identify novel genes associated with Menière's disease. *PLoS One*, *12*(8), e0182592.
- Li, S., Tang, C., Liu, X., Liu, Y., & Chen, J. (2019). Dual graph regularized compact feature representation for unsupervised feature selection. *Neurocomputing*, *331*, 77-96.
- Li, W., Chen, L., Zhao, J., & Wang, W. (2021). Embedded feature selection based on relevance vector machines with an approximated marginal likelihood and its industrial application. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.
- Li, Y., & Chen, L. (2014). Big biological data: challenges and opportunities. *Genomics, proteomics & bioinformatics*, *12*(5), 187.
- Li, Y., He, K., Kloster, K., Bindel, D., & Hopcroft, J. (2018). Local spectral clustering for overlapping community detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *12*(2), 1-27.
- Li, Y., Li, G., Yang, Y., Liang, X., & Xu, M. (2018). A fault diagnosis scheme for planetary gearboxes using adaptive multi-scale morphology filter and modified hierarchical permutation entropy. *Mechanical Systems and Signal Processing*, *105*, 319-337.
- Lin, D., Wong, R. C. W., Xie, M., & Wei, V. J. (2020, April). Index-free approach with theoretical guarantee for efficient random walk with restart query. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)* (pp. 913-924). IEEE.
- Liu, B. H. (2018). Differential Coexpression network analysis for gene expression data. In *Computational systems biology* (pp. 155-165). Humana Press, New York, NY.

- Liu, H., & Motoda, H. (Eds.). (2007). *Computational methods of feature selection*. CRC Press.
- Liu, K., Liu, H., Sun, D., & Zhang, L. (2021). Network Inference from Gene Expression Data with Distance Correlation and Network Topology Centrality. *Algorithms*, *14*(2), 61.
- Liu, M., & Zhang, D. (2016). Feature selection with effective distance. *Neurocomputing*, *215*, 100-109.
- Liu, X., Hong, Z., Liu, J., Lin, Y., Rodríguez-Patón, A., Zou, Q., & Zeng, X. (2020). Computational methods for identifying the critical nodes in biological networks. *Briefings in bioinformatics*, *21*(2), 486-497.
- Liu, X., Sang, X., Chang, J., Zheng, Y., & Han, Y. (2021). The water supply association analysis method in Shenzhen based on kmeans clustering discretization and apriori algorithm. *PloS one*, *16*(8), e0255684.
- Lokeswari, Y. V., & Jacob, S. G. (2017). Prediction of child tumours from microarray gene expression data through parallel gene selection and classification on spark. In *Computational Intelligence in Data Mining* (pp. 651-661). Springer, Singapore.
- Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome biology*, *15*(12), 1-21.
- Lu, S. J., Xie, J., Li, Y., Yu, B., Ma, Q., & Liu, B. Q. (2019). Identification of lncRNAs-gene interactions in transcription regulation based on co-expression analysis of RNA-seq data. *Math. Biosci. Eng*, *16*, 7112-7125.
- Luo, P., Chen, B., Liao, B., & Wu, F. X. (2021). Predicting disease-associated genes: Computational methods, databases, and evaluations. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *11*(2), e1383.



- Mafarja, M., & Mirjalili, S. (2018). Whale optimization approaches for wrapper feature selection. *Applied Soft Computing*, 62, 441-453.
- Mahmood, S., Shahbaz, M., & Guergachi, A. (2014). Negative and positive association rules mining from text using frequent and infrequent itemsets. *The Scientific World Journal*, 2014.
- Mahmud, M., Kaiser, M. S., McGinnity, T. M., & Hussain, A. (2021). Deep learning in mining biological data. *Cognitive Computation*, 13(1), 1-33.
- Maldonado, S., & López, J. (2018). Dealing with high-dimensional class-imbalanced datasets: Embedded feature selection for SVM classification. *Applied Soft Computing*, 67, 94-105.
- Malekipirbazari, M., Aksakalli, V., Shafqat, W., & Eberhard, A. (2021). Performance comparison of feature selection and extraction methods with random instance selection. *Expert Systems with Applications*, 179, 115072.
- Mallick, P. K., Satapathy, S. K., Mishra, S., Panda, A. R., & Mishra, D. (2021). Feature selection and classification for microarray data using ACO-FLANN framework. In *Intelligent and cloud computing* (pp. 491-501). Springer, Singapore.
- Mallik, S., & Zhao, Z. (2020). Detecting methylation signatures in neurodegenerative disease by density-based clustering of applications with reducing noise. *Scientific reports*, 10(1), 1-14.
- Mamoshina, P., Kochetov, K., Putin, E., Cortese, F., Aliper, A., Lee, W. S., ... & Zhavoronkov, A. (2018). Population specific biomarkers of human aging: a big data study using South Korean, Canadian, and Eastern European patient populations. *The Journals of Gerontology: Series A*, 73(11), 1482-1490.

- Mandal, K., & Sarmah, R. (2018). A density-based clustering for gene expression data using gene ontology. In *Proceedings of the International Conference on Computing and Communication Systems* (pp. 757-765). Springer, Singapore.
- Manikandan, G., & Abirami, S. (2021). Feature Selection Is Important: State-of-the-Art Methods and Application Domains of Feature Selection on High-Dimensional Data. In *Applications in Ubiquitous Computing* (pp. 177-196). Springer, Cham.
- Margaret H. Danham, and S. Sridhar (2006) "Data mining, Introductory and Advanced Topics", Person education, 1st ed., 2006.
- Matamala, N., Vargas, M. T., Gonzalez-Campora, R., Minambres, R., Arias, J. I., Menendez, P., ... & Benitez, J. (2015). Tumor microRNA expression profiling identifies circulating microRNAs for early breast cancer detection. *Clinical chemistry*, 61(8), 1098-1106.
- Mazumder, D. H., & Veilumuthu, R. (2019). An enhanced feature selection filter for classification of microarray cancer data. *ETRI Journal*, 41(3), 358-370.
- Mehmood, A., Afsar, K., Zameer, A., Awan, S. E., & Raja, M. A. Z. (2019). Integrated intelligent computing paradigm for the dynamics of micropolar fluid flow with heat transfer in a permeable walled channel. *Applied Soft Computing*, 79, 139-162.
- Mehmood, A., Sajjad, I. A., Ullah, M. N., Liaqat, R., Abbas, M. Z., & Wasaya, A. (2021, September). Evaluation of Feature Selection Methods for Non-Intrusive Load Monitoring. In *2021 International Conference on Technology and Policy in Energy and Electric Power (ICT-PEP)* (pp. 324-329). IEEE.
- Miotto, R., Wang, F., Wang, S., Jiang, X., & Dudley, J. T. (2018). Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6), 1236-1246.

- Misra, S., & Ray, S. S. (2017). Finding optimum width of discretization for gene expressions using functional annotations. *Computers in biology and medicine*, 90, 59-67.
- Miswan, N. H., Chan, C. S., & Ng, C. G. (2021). Association Rules Mining for Hospital Readmission: A Case Study. *Mathematics*, 9(21), 2706.
- Mohamed, R., & Samsudin, N. A. A (2021) New Discretization Approach of Bat and K-Means. *International Journal of Advanced Computer Science and Applications*, Vol. 12, No. 1, 2021
- Mohammed, S. A., Darrab, S., Noaman, S. A., & Saake, G. (2020, July). Analysis of breast cancer detection using different machine learning techniques. In *International Conference on Data Mining and Big Data* (pp. 108-117). Springer, Singapore.
- Moradi, P., & Rostami, M. (2015). Integration of graph clustering with ant colony optimization for feature selection. *Knowledge-Based Systems*, 84, 144-161.
- Morovvat, M., & Osareh, A. (2016). An ensemble of filters and wrappers for microarray data classification. *Mach. Learn. Appl. An Int. J*, 3(2), 1-17.
- Moslehi, F., & Haeri, A. (2021). A novel feature selection approach based on clustering algorithm. *Journal of Statistical Computation and Simulation*, 91(3), 581-604.
- Mu, Y., Liu, X., & Wang, L. (2018). A Pearson's correlation coefficient-based decision tree and its parallel implementation. *Information Sciences*, 435, 40-58.
- Mukras, R., Wiratunga, N., Lothian, R., Chakraborti, S., & Harper, D. (2007). Information gain feature selection for ordinal text classification using probability re-distribution. In *Proceedings of the Textlink workshop at IJCAI* (Vol. 7, p. 16).

- Musich, R., Cadle-Davidson, L., & Osier, M. V. (2021). Comparison of short-read sequence aligners indicates strengths and weaknesses for biologists to consider. *Frontiers in Plant Science*, *12*.
- Nagarajan, R., Miller, C. S., Dawson III, D., & Ebersole, J. L. (2019). Biologic modelling of periodontal disease progression. *Journal of clinical periodontology*, *46*(2), 160-169.
- Nagata, K., Washio, T., Kawahara, Y., & Unami, A. (2014). Toxicity prediction from toxicogenomic data based on class association rule mining. *Toxicology reports*, *1*, 1133-1142.
- Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of big data*, *2*(1), 1-21.
- Nguyen, L. H., & Holmes, S. (2019). Ten quick tips for effective dimensionality reduction. *PLoS computational biology*, *15*(6), e1006907.
- Nia, A. M., Chen, T., Barnette, B. L., Khanipov, K., Ullrich, R. L., Bhavnani, S. K., & Emmett, M. R. (2020). Efficient identification of multiple pathways: RNA-Seq analysis of livers from <sup>56</sup>Fe ion irradiated mice. *BMC bioinformatics*, *21*(1), 1-12.
- Nimbalkar, P., & Kshirsagar, D. (2021). Feature selection for intrusion detection system in internet-of-things (iot). *ICT Express*, *7*(2), 177-181.
- Noor, M. B. T., Zenia, N. Z., Kaiser, M. S., Mahmud, M., & Mamun, S. A. (2019, December). Detecting neurodegenerative disease from MRI: a brief review on a deep learning perspective. In International conference on brain informatics (pp. 115-125). Springer, Cham.

- Ovens, K., Maleki, F., Eames, B. F., & McQuillan, I. (2021). Juxtapose: a gene-embedding approach for comparing co-expression networks. *BMC bioinformatics*, 22(1), 1-26.
- Overend, G., Luo, Y., Henderson, L., Douglas, A. E., Davies, S. A., & Dow, J. A. (2016). Molecular mechanism and functional significance of acid generation in the *Drosophila* midgut. *Scientific reports*, 6(1), 1-11.
- Patel, S. P., & Upadhyay, S. H. (2020). Euclidean distance based feature ranking and subset selection for bearing fault diagnosis. *Expert Systems with Applications*, 154, 113400.
- Pati, J. (2018). Gene expression analysis for early lung cancer prediction using machine learning techniques: An eco-genomics approach. *IEEE Access*, 7, 4232-4238.
- Patro, R., Duggal, G., Love, M. I., Irizarry, R. A., & Kingsford, C. (2017). Salmon provides fast and bias-aware quantification of transcript expression. *Nature methods*, 14(4), 417-419.
- Peng, Z., Daraei, R., Ahmadpanahi, S. M., Danesh, A. S., Siadat, S., Pirozmand, P., & Oskouei, R. J. (2021). Proposing a density-based clustering approach (DBCA) to aggregate data collected from the environment in arid area for desertification. *Wireless Communications and Mobile Computing*, 2021.
- Pérez-Rubio, P., Lottaz, C., & Engelmann, J. C. (2019). FastqPuri: high-performance preprocessing of RNA-seq data. *BMC bioinformatics*, 20(1), 226.
- Petrillo, U. F., Sorella, M., Cattaneo, G., Giancarlo, R., & Rombo, S. E. (2019). Analyzing big datasets of genomic sequences: fast and scalable collection of k-mer statistics. *BMC bioinformatics*, 20(4), 1-14.

- Putin, E., Mamoshina, P., Aliper, A., Korzinkin, M., Moskalev, A., Kolosov, A., ... & Zhavoronkov, A. (2016). Deep biomarkers of human aging: application of deep neural networks to biomarker development. *Aging (Albany NY)*, 8(5), 1021.
- Pyrkov, T. V., Slipensky, K., Barg, M., Kondrashin, A., Zhurov, B., Zenin, A., ... & Fedichev, P. O. (2018). Extracting biological age from biomedical data via deep learning: too much of a good thing?. *Scientific reports*, 8(1), 1-11.
- Quang, D., & Xie, X. (2016). DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic acids research*, 44(11), e107-e107.
- Quintana, D. S., Rokicki, J., van der Meer, D., Alnæs, D., Kaufmann, T., Córdova-Palomera, A., ... & Westlye, L. T. (2019). Oxytocin pathway gene networks in the human brain. *Nature communications*, 10(1), 1-12.
- Rahman, F., & Mahmood, A. (2022). A Comprehensive Analysis of Most Relevant Features Causes Heart Disease Using Machine Learning Algorithms. In *Proceedings of the International Conference on Big Data, IoT, and Machine Learning* (pp. 63-73). Springer, Singapore.
- Rahman, S. A., & Adjeroh, D. A. (2019). Deep learning using convolutional LSTM estimates biological age from physical activity. *Scientific reports*, 9(1), 1-15.
- Ranjan, R., & Sharma, A. (2019). Evaluation of frequent itemset mining platforms using apriori and fp-growth algorithm. *International Journal of Information Systems & Management Science*, 2(2).
- Rao, R. R., & Makkithaya, K. (2017). Learning from a class imbalanced public health dataset: A cost-based comparison of classifier performance. *International Journal of Electrical and Computer Engineering*, 7(4), 2215.

- Rathor, A., & Gyanchandani, M. (2017, December). A review at Machine Learning algorithms targeting big data challenges. In *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)* (pp. 1-7). IEEE.
- Raunak, V. (2017). Simple and effective dimensionality reduction for word embeddings. *arXiv preprint arXiv:1708.03629*.
- Rawat, A. S., Rana, A., Kumar, A., & Bagwari, A. (2018). Application of multi layer artificial neural network in the diagnosis system: a systematic review. *IAES International Journal of Artificial Intelligence*, 7(3), 138.
- Ray, P., Reddy, S. S., & Banerjee, T. (2021). Various dimension reduction techniques for high dimensional data analysis: a review. *Artificial Intelligence Review*, 54(5), 3473-3515
- Ray, R. B., Kumar, M., & Rath, S. K. (2016, April). Fast in-memory cluster computing of sizeable microarray using spark. In *2016 International Conference on Recent Trends in Information Technology (ICRTIT)* (pp. 1-6). IEEE.
- Reddy, C. K., & Vinzamuri, B. (2018). A survey of partitional and hierarchical clustering algorithms. In *Data clustering* (pp. 87-110). Chapman and Hall/CRC.
- Reddy, G. T., Reddy, M. P. K., Lakshmana, K., Kaluri, R., Rajput, D. S., Srivastava, G., & Baker, T. (2020). Analysis of dimensionality reduction techniques on big data. *IEEE Access*, 8, 54776-54788.
- Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. *Encyclopedia of database systems*, 5, 532-538.
- Ren, Y., Wang, N., Li, M., & Xu, Z. (2020). Deep density-based image clustering. *Knowledge-Based Systems*, 197, 105841.

- Riquelme Medina, I., & Lubovac-Pilav, Z. (2016). Gene co-expression network analysis for identifying modules and functionally enriched pathways in type 1 diabetes. *PloS one*, *11*(6), e0156006.
- Rosone, G., & Sciortino, M. (2013, July). The Burrows-Wheeler transform between data compression and combinatorics on words. In *Conference on Computability in Europe* (pp. 353-364). Springer, Berlin, Heidelberg.
- Roy, S. S., & Taguchi, Y. H. (2021). Identification of genes associated with altered gene expression and m6A profiles during hypoxia using tensor decomposition based unsupervised feature extraction. *Scientific reports*, *11*(1), 1-18.
- Rtayli, N., & Enneya, N. (2020). Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization. *Journal of Information Security and Applications*, *55*, 102596.
- Ryan, C. (2016). *Occupant location prediction in smart buildings using association rule mining* (Doctoral dissertation, University College Cork).
- Sadiq, A., Yahya, N., & Tang, T. B. (2021, December). Diagnosis of Alzheimer's Disease Using Pearson's Correlation and ReliefF Feature Selection Approach. In *2021 International Conference on Decision Aid Sciences and Application (DASA)* (pp. 578-582). IEEE.
- Sagin, A. N., & Ayvaz, B. (2018). Determination of association rules with market basket analysis: application in the retail sector. *Southeast Europe Journal of Soft Computing*, *7*(1).
- Saheed, Y. K., & Hambali, M. A. (2021, October). Customer Churn Prediction in Telecom Sector with Machine Learning and Information Gain Filter Feature Selection Algorithms. In *2021 International Conference on Data Analytics for Business and Industry (ICDABI)* (pp. 208-213). IEEE.



- Şahin, D. Ö., Kural, O. E., Akleyek, S., & Kılıç, E. (2021). A novel Android malware detection system: adaption of filter-based feature selection methods. *Journal of Ambient Intelligence and Humanized Computing*, 1-15.
- Salekin, A., & Stankovic, J. (2016, October). Detection of chronic kidney disease and selecting important predictive attributes. In *2016 IEEE International Conference on Healthcare Informatics (ICHI)* (pp. 262-270). IEEE.
- Salem, H., Attiya, G., & El-Fishawy, N. (2017). Classification of human cancer diseases by gene expression profiles. *Applied Soft Computing*, 50, 124-134.
- Salikhov, K. (2017). *Efficient algorithms and data structures for indexing DNA sequence data*. PhD 773 (Doctoral dissertation, Thesis, Université Paris-Est).
- Salo, F., Nassif, A. B., & Essex, A. (2019). Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection. *Computer Networks*, 148, 164-175.
- Samanta, S., Pal, M., Mahapatra, R., Das, K., & Bhadoria, R. S. (2021). A study on semi-directed graphs for social media networks. *International Journal of Computational Intelligence Systems*, 14(1), 1034.
- Saraswathi, V., & Gupta, D. (2019, January). Classification of Brain Tumor using PCA-RF in MR Neurological Images. In *2019 11th International Conference on Communication Systems & Networks (COMSNETS)* (pp. 440-443). IEEE.
- Sari, D. P., Rosadi, D., Effendie, A. R., & Danardono, D. (2021). Discretization methods for Bayesian networks in the case of the earthquake. *Bulletin of Electrical Engineering and Informatics*, 10(1), 299-307.

- Sarkar, S., Pramanik, A., Maiti, J., & Reniers, G. (2020). Predicting and analyzing injury severity: A machine learning-based approach using class-imbalanced proactive and reactive data. *Safety science*, *125*, 104616.
- Sarraf, J., & Pattnaik, P. K. (2020). Brain–computer interfaces and their applications. In *An Industrial IoT Approach for Pharmaceutical Industry Growth* (pp. 31-54). Academic Press.
- Sawangarreerak, S., & Thanathamathsee, P. (2021). Detecting and analyzing fraudulent patterns of financial statement for open innovation using discretization and association rule mining. *Journal of Open Innovation: Technology, Market, and Complexity*, *7*(2), 128.
- Seo, H., & Cho, D. H. (2020). Cancer-related gene signature selection based on boosted regression for multilayer perceptron. *IEEE Access*, *8*, 64992-65004.
- Serra, A., Galdi, P., & Tagliaferri, R. (2018). Machine learning for bioinformatics and neuroimaging. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(5), e1248.
- Shabtay, L., Fournier-Viger, P., Yaari, R., & Dattner, I. (2021). A guided FP-Growth algorithm for mining multitude-targeted item-sets and class association rules in imbalanced data. *Information Sciences*, *553*, 353-375.
- Shah, F. P., & Patel, V. (2016, March). A review on feature selection and feature extraction for text classification. In *2016 international conference on wireless communications, signal processing and networking (WiSPNET)* (pp. 2264-2268). IEEE.
- Shahee, S. A., & Ananthakumar, U. (2020). An effective distance based feature selection approach for imbalanced data. *Applied Intelligence*, *50*(3), 717-745.

- Sharma, N., Chakrabarti, A., & Balas, V. E. (2019). Data management, analytics and innovation. *Proceedings of ICDMAI, 1*.
- Hammami, M., Bechikh, S., Hung, C. C., & Ben Said, L. (2019). A multi-objective hybrid filter-wrapper evolutionary approach for feature selection. *Memetic Computing, 11*(2), 193-208.
- Shekhovtsov, A., & Salabun, W. (2020). A comparative case study of the VIKOR and TOPSIS rankings similarity. *Procedia Computer Science, 176*, 3730-3740.
- Shin, K., Shin, W., Ha, J. W., & Kwon, S. (2021). Graphs, Entities, and Step Mixture for Enriching Graph Representation. *IEEE Access, 9*, 144025-144034.
- Shrestha, A. M. S., Frith, M. C., & Horton, P. (2014). A bioinformatician's guide to the forefront of suffix array construction algorithms. *Briefings in bioinformatics, 15*(2), 138-154.
- Shui, Y., & Cho, Y. R. (2016, December). Filtering association rules in Gene Ontology based on term specificity. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (pp. 1314-1321). IEEE.
- Singh, S. S., Mishra, S., Kumar, A., & Biswas, B. (2022). Link Prediction on Social Networks Based on Centrality Measures. In *Principles of Social Networking* (pp. 71-89). Springer, Singapore.
- Sridhar, S., & Sanagavarapu, S. (2021, November). DarkNet Traffic Classification Pipeline with Feature Selection and Conditional GAN-based Class Balancing. In *2021 IEEE 20th International Symposium on Network Computing and Applications (NCA)* (pp. 1-4). IEEE

- Sun, S., Zhu, J., Ma, Y., & Zhou, X. (2019). Accuracy, robustness and scalability of dimensionality reduction methods for single-cell RNA-seq analysis. *Genome biology*, 20(1), 1-21.
- Tabachnick, A. R., Valadez, E. A., Palmwood, E. N., Zajac, L., Simons, R. F., & Dozier, M. (2018). Depressive symptoms and error-related brain activity in CPS-referred children. *Psychophysiology*, 55(11), e13211.
- Tan, H., Wang, G., Wang, W., & Zhang, Z. (2020). Feature selection based on distance correlation: a filter algorithm. *Journal of Applied Statistics*, 1-16.
- Tanwani, A. K., Afridi, J., Shafiq, M. Z., & Farooq, M. (2009, April). Guidelines to select machine learning scheme for classification of biomedical datasets. In *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics* (pp. 128-139). Springer, Berlin, Heidelberg.
- Team, R. C. (2017). R: A language and environment for statistical computing.
- Telikani, A., Gandomi, A. H., & Shahbahrami, A. (2020). A survey of evolutionary computation for association rule mining. *Information Sciences*, 524, 318-352.
- Teng, H., Yuan, Y., & Bar-Joseph, Z. (2022). Clustering spatial transcriptomics data. *Bioinformatics*, 38(4), 997-1004.
- Thakkar, A., & Lohiya, R. (2021). Attack classification using feature selection techniques: a comparative study. *Journal of Ambient Intelligence and Humanized Computing*, 12(1), 1249-1266.
- Thrun, M. C., & Ultsch, A. (2021). Using projection-based clustering to find distance-and density-based clusters in high-dimensional data. *Journal of Classification*, 38(2), 280-312.

- Todorov, H., Fournier, D., & Gerber, S. (2018). Principal components analysis: theory and application to gene expression data analysis. *Genom. Comput. Biol*, 4(2).
- Toğaçar, M., Ergen, B., Cömert, Z., & Özyurt, F. (2020). A deep feature learning model for pneumonia detection applying a combination of mRMR feature selection and machine learning models. *Irbm*, 41(4), 212-222.
- Turgut, S., Dağtekin, M., & Ensari, T. (2018, April). Microarray breast cancer data classification using machine learning methods. In *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)* (pp. 1-3). IEEE.
- Urbanowicz, R. J., Meeker, M., La Cava, W., Olson, R. S., & Moore, J. H. (2018). Relief-based feature selection: Introduction and review. *Journal of biomedical informatics*, 85, 189-203.
- Valdeolivas, A., Tichit, L., Navarro, C., Perrin, S., Odelin, G., Levy, N., ... & Baudot, A. (2019). Random walk with restart on multiplex and heterogeneous biological networks. *Bioinformatics*, 35(3), 497-505.
- Vanitha, C. D. A., Devaraj, D., & Venkatesulu, M. (2015). Gene expression data classification using support vector machine and mutual information-based gene selection. *procedia computer science*, 47, 13-21.
- Vapnik, V. (1998). *Statistical learning theory* new york. NY: Wiley, 1, 2.
- Viktoratos, I., Tsadiras, A., & Bassiliades, N. (2018). Combining community-based knowledge with association rule mining to alleviate the cold start problem in context-aware recommender systems. *Expert systems with applications*, 101, 78-90.
- Von Der Weid, B., Rossier, D., Lindup, M., Tuberosa, J., Widmer, A., Dal Col, J., ... & Rodriguez, I. (2015). Large-scale transcriptional profiling of chemosensory

neurons identifies receptor-ligand pairs in vivo. *Nature Neuroscience*, 18(10), 1455-1463.

Vural, H., Kaya, M., & Alhadj, R. (2019, August). A model based on random walk with restart to predict circRNA-disease associations on heterogeneous network. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (pp. 929-932).

Wang, B., Mezlini, A. M., Demir, F., Fiume, M., Tu, Z., Brudno, M., ... & Goldenberg, A. (2014). Similarity network fusion for aggregating data types on a genomic scale. *Nature methods*, 11(3), 333-337.

Wang, C., Long, Y., Li, W., Dai, W., Xie, S., Liu, Y., ... & Duan, Y. (2020). Exploratory study on classification of lung cancer subtypes through a combined K-nearest neighbor classifier in breathomics. *Scientific reports*, 10(1), 1-12.

Wang, J., Chen, S., Dong, L., & Wang, G. (2021). CHTKC: a robust and efficient k-mer counting algorithm based on a lock-free chaining hash table. *Briefings in Bioinformatics*, 22(3), bbaa063.

Wang, J., Wang, K., Niu, J., & Liu, W. (2018, January). A K-medoids based clustering algorithm for wireless sensor networks. In *2018 international workshop on advanced image technology (IWAIT)* (pp. 1-4). IEEE.

Wang, K., & Fu, X. (2017, May). Research on centrality of urban transport network nodes. In *AIP conference proceedings* (Vol. 1839, No. 1, p. 020181). AIP Publishing LLC.

Wang, L., Xiao, Y., Li, J., Feng, X., Li, Q., & Yang, J. (2019). IIRWR: internal inclined random walk with restart for LncRNA-disease association prediction. *IEEE Access*, 7, 54034-54041.

- Wang, Z., Guo, L., Wang, S., Chen, L., & Wang, H. (2019). Review of random walk in image processing. *Archives of Computational Methods in Engineering*, 26(1), 17-34.
- Wang, Z., Zhao, C., Wang, Y., Sun, Z., & Wang, N. (2018). PANDA: Protein function prediction using domain architecture and affinity propagation. *Scientific reports*, 8(1), 1-10.
- Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of ‘small-world’ networks. *nature*, 393(6684), 440-442.
- Waylen, L. N., Nim, H. T., Martelotto, L. G., & Ramialison, M. (2020). From whole-mount to single-cell spatial assessment of gene expression in 3D. *Communications biology*, 3(1), 1-11.
- Wei, Y., & Sekiya, Y. (2021, November). Feature Selection Approach for Phishing Detection Based on Machine Learning. In *International Conference on Applied CyberSecurity* (pp. 61-70). Springer, Cham.
- Wekesa, J. S., Meng, J., & Luan, Y. (2020). A deep learning model for plant lncRNA-protein interaction prediction with graph attention. *Molecular Genetics and Genomics*, 295(5), 1091-1102.
- Wen, F., Zhang, G., Sun, L., Wang, X., & Xu, X. (2019). A hybrid temporal association rules mining method for traffic congestion prediction. *Computers & Industrial Engineering*, 130, 779-787.
- Whigham, P. A., & Spencer, H. G. (2021). Graph-structured populations and the Hill–Robertson effect. *Royal Society open science*, 8(3), 201831.

- Wicaksono, D., Jambak, M. I., & Saputra, D. M. (2020). The comparison of apriori algorithm with preprocessing and FP-growth algorithm for finding frequent data pattern in association rule. *vol, 172*, 315-319.
- Williams, C. R., Baccarella, A., Parrish, J. Z., & Kim, C. C. (2016). Trimming of sequence reads alters RNA-Seq gene expression estimates. *BMC bioinformatics, 17*(1), 1-13.
- Wu, G., & Xu, J. (2015, October). Optimized approach of feature selection based on information gain. In *2015 International Conference on Computer Science and Mechanical Automation (CSMA)* (pp. 157-161). IEEE.
- Wu, P. Y., Cheng, C. W., Kaddi, C. D., Venugopalan, J., Hoffman, R., & Wang, M. D. (2016). –Omic and electronic health record big data analytics for precision medicine. *IEEE Transactions on Biomedical Engineering, 64*(2), 263-273.
- Wu, Y., Di, L., Ming, Y., Lv, H., & Tan, H. (2019). High-resolution optical remote sensing image registration via reweighted random walk based hyper-graph matching. *Remote Sensing, 11*(23), 2841.
- Wu, Y., Lao, B., Ma, X., & Nong, G. (2019, December). An improved algorithm for building suffix array in external memory. In *International Symposium on Parallel Architectures, Algorithms and Programming* (pp. 320-330). Springer, Singapore.
- Xia, F., Liu, H., Lee, I., & Cao, L. (2016). Scientific article recommendation: Exploiting common author relations and historical preferences. *IEEE Transactions on Big Data, 2*(2), 101-112.
- Xiong, Y., Wang, K., Zhou, H., Peng, L., You, W., & Fu, Z. (2018). Profiles of immune infiltration in colorectal cancer and their clinical significant: A gene expression-based study. *Cancer medicine, 7*(9), 4496-4508.



- Xu, C., & Jackson, S. A. (2019). Machine learning and complex biological data. *Genome Biology*, 20(1), 1-4.
- Xu, J., Mu, H., Wang, Y., & Huang, F. (2018). Feature genes selection using supervised locally linear embedding and correlation coefficient for microarray classification. *Computational and mathematical methods in medicine*, 2018.
- Xu, R., & Wunsch, D. C. (2009). II, Clustering. Hoboken. NJ: Wiley/IEEE Press, 6, 583-617.
- Xu, S., Cui, Y., Yang, C., Wei, S., Dong, W., Huang, L., ... & Wang, W. (2021). The fuzzy comprehensive evaluation (FCE) and the principal component analysis (PCA) model simulation and its applications in water quality assessment of Nansi Lake Basin, China. *Environmental Engineering Research*, 26(2), 222-232.
- Yan, H., Zhang, Q., Mao, D., Lu, Z., Guo, D., & Chen, S. (2021, July). Anomaly Detection of Network Streams via Dense Subgraph Discovery. In *2021 International Conference on Computer Communications and Networks (ICCCN)* (pp. 1-9). IEEE.
- Yan, X., & Jia, M. (2019). Intelligent fault diagnosis of rotating machinery using improved multiscale dispersion entropy and mRMR feature selection. *Knowledge-Based Systems*, 163, 450-471.
- Yang, F., Wang, Z., Xiao, J., & Satoh, S. I. (2020, April). Mining on heterogeneous manifolds for zero-shot cross-modal image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 07, pp. 12589-12596).
- Yang, W., Si, Y., Wang, D., & Guo, B. (2018). Automatic recognition of arrhythmia based on principal component analysis network and linear support vector machine. *Computers in biology and medicine*, 101, 22-32.

- Yegnanarayanan, V. (2020). Understanding Alzheimer's Disease through Graph Theory. *Journal of Applied Mathematics and Physics*, 8(10), 2182.
- Yildirim, M. A., & Coscia, M. (2014). Using random walks to generate associations between objects. *PloS one*, 9(8), e104813.
- Yousef, M., Kumar, A., & Bakir-Gungor, B. (2021). Application of biological domain knowledge based feature selection on gene expression data. *Entropy*, 23(1), 2.
- Yu, G., Zhu, H., & Domeniconi, C. (2015). Predicting protein functions using incomplete hierarchical labels. *BMC bioinformatics*, 16(1), 1-12.
- Yu, H., Kim, P. M., Sprecher, E., Trifonov, V., & Gerstein, M. (2007). The importance of bottlenecks in protein networks: correlation with gene essentiality and expression dynamics. *PLoS Comput Biol*, 3(4), e59.
- Yu, X., Yu, G., & Wang, J. (2017). Clustering cancer gene expression data by projective clustering ensemble. *PloS one*, 12(2), e0171429.
- Yu, X., Yu, G., & Wang, J. (2017). Clustering cancer gene expression data by projective clustering ensemble. *PloS one*, 12(2), e0171429.
- Yuan, M., Yang, Z., Huang, G., & Ji, G. (2017). Feature selection by maximizing correlation information for integrated high-dimensional protein data. *Pattern Recognition Letters*, 92, 17-24.
- Zaffar, M., Hashmani, M. A., & Savita, K. S. (2017, November). Performance analysis of feature selection algorithm for educational data mining. In *2017 IEEE Conference on Big Data and Analytics (ICBDA)* (pp. 7-12). IEEE.
- Zahiri, J., Hannon Bozorgmehr, J., & Masoudi-Nejad, A. (2013). Computational prediction of protein-protein interaction networks: algorithms and resources. *Current genomics*, 14(6), 397-414.

- Zakaria, W., Kotb, Y., & Ghaleb, F. (2014). MCR-Miner:: Maximal confident association rules miner algorithm for up/down-expressed genes. *Applied Mathematics & Information Sciences*, 8(2), 799.
- Zaki, M. J. (2000). Scalable algorithms for association mining. *IEEE transactions on knowledge and data engineering*, 12(3), 372-390.
- Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020). A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *Journal of Applied Science and Technology Trends*, 1(2), 56-70.
- Zhang, B., & Horvath, S. (2005). A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1).
- Zhang, S. W., Shao, D. D., Zhang, S. Y., & Wang, Y. B. (2014). Prioritization of candidate disease genes by enlarging the seed set and fusing information of the network topology and gene expression. *Molecular BioSystems*, 10(6), 1400-1408.
- Zhang, S., Zhou, J., Hu, H., Gong, H., Chen, L., Cheng, C., & Zeng, J. (2016). A deep learning framework for modeling structural features of RNA-binding protein targets. *Nucleic acids research*, 44(4), e32-e32.
- Zhang, X., & Zeng, X. (2019). Integrative approaches for predicting microRNA function and prioritizing disease-related microRNA using biological interaction networks. *Bio-inspired Computing Models And Algorithms*, 75-105.
- Zhang, Y., Jiang, Y., Qi, L., Bhuiyan, M. Z. A., & Qian, P. (2021). Epilepsy diagnosis using multi-view & multi-medoid entropy-based clustering with privacy protection. *ACM Transactions on Internet Technology*, 21(2), 1-21.
- Zhao, H., Liu, H., Xu, J., & Deng, W. (2019). Performance prediction using high-order differential mathematical morphology gradient spectrum entropy and extreme

- learning machine. *IEEE Transactions on Instrumentation and Measurement*, 69(7), 4165-4172.
- Zhavoronkov, A., & Mamoshina, P. (2019). Deep aging clocks: the emergence of AI-based biomarkers of aging and longevity. *Trends in Pharmacological Sciences*, 40(8), 546-549.
- Zheng, A., & Casari, A. (2018). *Feature engineering for machine learning: principles and techniques for data scientists*. " O'Reilly Media, Inc."
- Zhou, P., Chen, J., Du, L., & Li, X. (2022). Balanced Spectral Feature Selection. *IEEE Transactions on Cybernetics*.
- Zhu, P., Zhu, W., Hu, Q., Zhang, C., & Zuo, W. (2017). Subspace clustering guided unsupervised feature selection. *Pattern Recognition*, 66, 364-374
- Zhu, T., Lin, Y., & Liu, Y. (2017). Synthetic minority oversampling technique for multiclass imbalance problems. *Pattern Recognition*, 72, 327-340.
- Zhu, X., Zhang, S., Li, Y., Zhang, J., Yang, L., & Fang, Y. (2018). Low-rank sparse subspace for spectral clustering. *IEEE Transactions on Knowledge and Data Engineering*, 31(8), 1532-1543.
- Ziemann, M., Kaspi, A., & El-Osta, A. (2016). Evaluation of microRNA alignment techniques. *Rna*, 22(8), 1120-1138.
- Zong, L., Zhang, X., Zhao, L., Yu, H., & Zhao, Q. (2017). Multi-view clustering via multi-manifold regularized non-negative matrix factorization. *Neural Networks*, 88, 74-89.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

- Daoud, M., & Mayo, M. (2019). A survey of neural network-based cancer prediction models from microarray data. *Artificial intelligence in medicine*, 97, 204-214.
- Basavegowda, H. S., & Dagnev, G. (2020). Deep learning approach for microarray cancer data classification. *CAAI Transactions on Intelligence Technology*, 5(1), 22-33.
- Guo, W., Xu, Y., & Feng, X. (2017). DeepMetabolism: a deep learning system to predict phenotype from genome sequencing. *arXiv preprint arXiv:1705.03094*.
- Fakoor, R., Ladhak, F., Nazi, A., & Huber, M. (2013, June). Using deep learning to enhance cancer diagnosis and classification. *In Proceedings of the international conference on machine learning* (Vol. 28, pp. 3937-3949). ACM, New York, USA.
- Dincer, A. B., Celik, S., Hiranuma, N., & Lee, S. I. (2018). DeepProfile: Deep learning of cancer molecular profiles for precision medicine. *BioRxiv*, 278739.
- Samek, W., Montavon, G., Lapuschkin, S., Anders, C. J., & Müller, K. R. (2020). Toward interpretable machine learning: Transparent deep neural networks and beyond.
- Chakraborty, S., Tomsett, R., Raghavendra, R., Harborne, D., Alzantot, M., Cerutti, F., ... & Gurram, P. (2017, August). Interpretability of deep learning models: A survey of results. In *2017 IEEE smartworld, ubiquitous intelligence & computing, advanced & trusted computed, scalable computing & communications, cloud & big data computing, Internet of people and smart city innovation* (pp. 1-6). IEEE.
- Guidotti, R., Monreale, A., Turini, F., Pedreschi, D., & Giannotti, F. (2018). A survey of methods for explaining black box models. *arXiv preprint arXiv:1802.01933*.

- Danaee, P., Ghaeini, R., & Hendrix, D. A. (2017). A deep learning approach for cancer detection and relevant gene identification. In Pacific symposium on biocomputing 2017 (pp. 219-229).
- Teixeira, V., Camacho, R., & Ferreira, P. G. (2017, November). Learning influential genes on cancer gene expression data with stacked denoising autoencoders. In 2017 IEEE *International Conference on Bioinformatics and Biomedicine (BIBM)* (pp. 1201-1205). IEEE.
- Way, G. P., & Greene, C. S. (2018). Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2018: Proceedings of the Pacific Symposium* (pp. 80-91).
- Sharifi-Noghabi, H., Liu, Y., Erho, N., Shrestha, R., Alshalalfa, M., Davicioni, E., ... & Ester, M. (2019). Deep genomic signature for early metastasis prediction in prostate cancer. *BioRxiv*, 276055.
- Hanczar, B., Henriette, M., Ratovomanana, T., & Zehraoui, F. (2018). Phenotypes prediction from gene expression data with deep multilayer perceptron and unsupervised pre-training. *Int J Biosci Biochem Bioinform*, 8, 125-131.
- Weighill, D., Jones, P., Bleker, C., Ranjan, P., Shah, M., Zhao, N., ... & Jacobson, D. (2019). Multi-phenotype association decomposition: unraveling complex gene-phenotype relationships. *Frontiers in genetics*, 10, 417.

## APPENDICES

### Appendix I: Author's Publications during PhD Study

Gakii, C., Bwana, B. K., Mugambi, G. G., Mukoya, E., Mireji, P. O., & Rimiru, R. (2021). In silico-driven analysis of the *Glossina morsitans morsitans* antennae transcriptome in response to repellent or attractant compounds. *PeerJ*, 9, e11691.

Gakii, C., & Rimiru, R. (2021). Identification of cancer related genes using feature selection and association rule mining. *Informatics in Medicine Unlocked-Elsevier*, 24, 100595.

Gakii, C., Mireji, P. O., & Rimiru, R. (2022). Graph Based Feature Selection for Reduction of Dimensionality in Next-Generation RNA Sequencing Datasets. *Algorithms*, 15(1), 21.