Statistical and Deep Learning Methods for Cancer Genomic Data

UNIVERSITY OF
KWAZULU - NATAL

INYUVESI
YAKWAZULU-NATALI

Mohanad M. A. Mohammed

October, 2021

Statistical and Deep Learning Methods for Cancer Genomic Data

by

Mohanad M. A. Mohammed

A thesis submitted to the
University of KwaZulu-Natal
in fulfilment of the requirements for the degree
of
DOCTOR OF PHILOSOPHY
in
BIOSTATISTICS

Thesis Supervisors:
Prof. Henry G. Mwambi & Prof. Bernard O. Omolo

UNIVERSITY OF KWAZULU-NATAL
SCHOOL OF MATHEMATICS, STATISTICS AND COMPUTER SCIENCE
PIETERMARITZBURG CAMPUS, SOUTH AFRICA

Declaration

I, Mohanad M. A. Mohammed, declare that;

1. The research reported in this thesis, except where otherwise indicated, is my original research.
2. This thesis has not been submitted for any degree or examination at any other university.
3. This thesis does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.
4. This thesis does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then;
   (a) their words have been re-written but the general information attributed to them has been referenced, or
   (b) where their exact words have been used, then their writing has been placed in italics and referenced.
5. This thesis does not contain text, graphics or tables copied and pasted from the internet, unless specifically acknowledged, and the source being detailed in the thesis and in the reference sections.

[redacted] 07/06/2022

Mohanad M. A. Mohammed (Student) Date

[redacted] 15 June 2022

Prof. Henry G. Mwambi (Supervisor) Date

[redacted] 09/06/2022

Prof. Bernard O. Omolo (Co-Supervisor) Date

Disclaimer

This document describes work undertaken as a PhD programme of study at the University of KwaZulu-Natal (UKZN). All views and opinions expressed therein remain the sole responsibility of the author, and do not necessarily represent those of the institution.

Contents

List of Tables

List of Figures

Dedication

This dissertation is dedicated to my supervisors, my parents, my brothers, my sisters, my wife, teachers, and friends who have always taught me how to see the vitality in myself, conquer fears and overcome challenges. I hope that I have made you proud.

Acknowledgements

In the name of Allah, the Most Gracious and the Most Merciful. All praises to Allah, the Cherisher and Sustainer of the world. I am highly grateful to the Almighty Allah for His unmeasurable Mercies that saw me through my studies. It is such a great honour for me and humbling at the same time because it has been a long journey characterized by long hours of hard work and sacrifice. It is with pleasure and gratitude to acknowledge those who rendered assistance through the lengthy process of my study. I sincerely appreciate all the big and small contributions from everybody who either directly or indirectly helped make my studies less stressful and a success.

This work was funded by GSK Africa Non-Communicable Disease Open Lab through the DELTAS Africa Sub-Saharan African Consortium for Advanced Biostatistics (SSACAB) Grant No. 107754/Z/15/Z-training programme. The views expressed in this publication are those of the author(s) and not necessarily those of GSK. It is through the financial support from GSK SSACAB that I was able to pursue my PhD studies. I am grateful to the SSACAB consortium for giving me this opportunity.

I am specifically grateful to my supervisors, Prof. Henry Mwambi and Prof. Bernard Omolo for their presence, inspiration, patience, dedication and their powerful guidance and help. I again thank them for their availability and readiness to read and correct all the many errors in my work, as well as the financial support. Much thanks for their quick responses whenever I consulted. Their help in and outside the research has been immense. I am deeply grateful for their willingness to

work with me. Thank you both for sharing your enormous statistical knowledge with me.

Many thanks go to my colleagues at the University of KwaZulu-Natal, School of Mathematics, Statistics, and Computer Science (Pietermaritzburg Campus) Ms. Christel Bernard for ensuring I have a comfortable working place environment; Ms. Jesca Batidzirai, and fellow PhD candidates; Ashenafi Argaw Yirga, Innocent Mboya, and Alexanda Kasyoki for their help, encouragement and assisting on some methods relevant to my research. Special thanks to all my colleagues and friends inside and outside South Africa, especially Faculty of Mathematical and Computer Sciences, University of Gezira, Sudan Dr. Murtada K. Elbashir for his spiritual, moral and enthusiastic support.

I would like to thank my parents, Mohammed and Entisar who have taught me the important things in life and have supported all my endeavours, for their unconditional love, encouragement, patience for being away when they needed me the most. I am grateful and thankful to my brothers and sisters, Modather, Mogtba, Zahrra, and Tanzeel for their caring of my parents. Many thanks to the rest of my extended family for your moral support.

Above all I would like to thank my wife Tehnan Mohamed for her love and constant support, for all the late nights and early mornings, and for keeping me sane over the past few months. Thank you for being my muse, editor, proof-reader, and sounding board. But most of all, thank you for being my best friend. I owe you everything. Last, but not least, thanks to everyone who his or her names may not appear here, but I won't forget their help. May the Almighty Allah bless them all.

List of publications

The contents of this thesis are based on the following publications:

1. **Mohanad Mohammed**, Henry Mwambi & Bernard Omolo, (2021). Colorectal Cancer Classification and Survival Analysis Based on an Integrated RNA and DNA Molecular Signature, Current Bioinformatics, 16(4). https://doi.org/10.2174/1574893615999200711170445

2. **Mohanad Mohammed**, Henry Mwambi, Innocent B. Mboya, Murtada K. Elbashir, & Bernard Omolo, (2021). A stacking ensemble deep learning approach to cancer type classification based on TCGA data. Scientific Reports, 11(1).

3. **Mohanad Mohammed**, Innocent B. Mboya, Henry Mwambi, Murtada K. Elbashir, & Bernard Omolo. (2021) Predictors of colorectal cancer survival using cox regression and random survival forests models based on gene expression data. PLoS ONE, 16(12).

The author has also contributed to the following publications during the course of his PhD training:

1. Elbashir, M. K., Ezz, M., **Mohammed, M.**, Saloum, S. S. (2019). Lightweight convolutional neural network for breast cancer classification using RNA-seq gene expression data. IEEE Access, 7, 185338-185348.

2. Mboya, I. B., Mahande, M. J., **Mohammed, M.**, Obure, J., & Mwambi, H. G. (2020). Prediction of perinatal death using machine learning models: a birth registry-based cohort study in northern Tanzania. BMJ open, 10(10), e040132.

3. Alharbi, F., Elbashir, M. K., **Mohammed, M.**, & Mustafa, M. E. (2020). Fine-Tuning Pre-Trained Convolutional Neural Networks for Women Common Cancer Classification using RNA-Seq Gene Expression. International Journal of Advanced Computer Science and Applications, 11(11).

Abbreviations

NCDs	Non-Communicable Diseases
HTS	High Throughput Sequencing
DNA	Deoxyribonucleic Acid
RNA	Ribonucleic Acid
mRNA	Messenger RNA
RNASeq	RNA Sequencing
cDNA	Complementary DNA
NGS	Next-Generation Sequencing
GANN	Gene Annotation
SAMANN	Sample Annotation
DGE	Differential Gene Expression
NBD	Negative Binomial Distributions
FWER	Family-Wise Error Rate
BH	Benjamini-Hochberg
LIMMA	Linear Models for Microarray Data
KNN	K-Nearest Neighbor
NB	Naive Bayes
SVM	Support Vector Machines
NNs	Neural Networks
LDA	Linear Discriminant Analysis
PLDA	Poisson Linear Discriminant Analysis
NBLDA	Negative Binomial Linear Discriminant Analysis
PCA	Principal Component Analysis
SVD	Singular Value Decomposition
ICA	Independent Component Analysis
ML	Machine Learning
OVA	one versus all
OVR	one versus rest
DL	Deep Learning

AI	Artificial Intelligence
CV	Cross-Validation
RFE	Recursive Feature Elimination
GA	Genetic Algorithm
LASSO	Least Absolute Shrinkage and Selection Operator
KM	Kaplan-Meier
MCAR	Missing Completely at Random
MAR	Missing at Random
MNAR	missing not at random
MI	Multiple Imputation
CM	Confusion Matrix
AUC	Area Under the Curve
BER	Balanced Error Rate
ROC	Receiver Operating Characteristic Curve
GEO	Gene Expression Omnibus
TCGA	The Cancer Genome Atlas
CNN	Convolutional Neural Network
CRC	Colorectal cancer
WHO	World Health Organization
ANN	Artificial Neural Networks
RF	Random Forests
ICA	Independent Component Analysis
IG	Information Gain
SGA	Standard Genetic Algorithm
CFS	Correlation-based Feature Selection
IBPSO	Improved-Binary Particle Swarm Optimization
bagSVM	Bagging SVM
CART	Classification and Regression Trees
mRMR	Minimum-Redundancy Maximum-Relevance
FFPE	Formalin-Fixed Paraffin-Embedded

CPM	Counts Per Million
GLM	Generalized Linear Model
MLE	Maximum Likelihood Estimate
DEGs	Differentially Expressed Genes
OS	Overall Survival
Cox PH	Cox Proportional Hazards
SHAP	Shapley Additive Explanations
HR	Hazard Ratio
IFS	Incremental Feature Selection
CIN	Chromosomal Instability
CPEM	Cancer Predictor using an Ensemble Model
LR	Logistics Regression
SVC	Support Vector Classifier
MLP	Multi-Layer Perceptron
DNN	Deep Neuronal Networks
1D-CNN	One-Dimensional Convolutional Neural Network
BRCA	BReast CAncer
COAD	COlon ADenocarcinoma
LUAD	LUng ADenocarcinoma
OV	Ovarian
THCA	THyroid CAncer
AAIC	Array-Array Intensity Correlation
SMOTE	Synthetic Minority Over-Sampling TEchnique
SVM-R	Support Vector Machine with Radial-basis function (RBF) kernel
SVM-L	Support Vector Machine with Linear Kernel
SVM-P	Support Vector Machine with Polynomial Kernel
ACC	Accuracy
CI	Confidence Interval
Kappa	Kappa Statistics
GEPIA	Gene Expression Profiling Interactive Analysis
RSF	Random Survival Forests

PH	Proportional Hazard
SD	Standard Deviation
FCS	Fully Conditional Specification
MCMC	Markov Chain Monte Carlo
OOB	Out-Of-Bag samples
CHF	Cumulative Hazard Function
BS	Brier Scores
IBS	Integrated Brier Scores
TPR	True Positive Rate
FPR	False Positive Rate
SNPs	Single Nucleotide Polymorphisms

Abstract

Statistical and machine learning methods have been applied in broad domains including the medical field. These methods have a massive impact on healthcare by providing the support for decision making to the specialist in diagnosis and prognosis of patient disease status and disease progression. Non-communicable diseases (NCDs) remain a major challenge the world over in the 21st century, especially in developing countries where resources are limited. Recent global public health research shows an epidemiological paradigm shift from infection to non-communicable diseases, which include cancer.

Cancer is considered the most devastating among all NCDs and is ranked second to malaria as the leading causes of death in the developing countries. Cancer occurs in many different types affecting all community members, where the general mechanism of cancer disease etiology is uncontrolled cells proliferation that leads to a malignant or cancerous tumor, and abnormalities at the molecular level. However, earlier detection and accurate diagnosis of cancer symptoms increase the probability of curing the condition, which has become the best strategy for fighting the disease. In the past few years, a vast amount of cancer data have been generated through new high throughput technologies. Traditional clinical and experimental approaches lack the capacity to handle such a massive scale of data. Therefore, computational methods have been introduced to biomedical investigations, including genes/biomarkers selection of cancer types and stages of the disease. Many computational tools have been developed based on different statistical and machine learning strategies and data science approaches.

We used statistical, machine and deep learning methods for cancer types, subtypes,

and survival prediction in this work. First, we developed a hybrid (DNA mutation and RNA expression) signature and assessed its predictive properties for colorectal cancer (CRC) patients' mutation status and survival. In addition, we proposed a stacking ensemble deep learning approach to evaluate and compare its predictive performance for cancer types (as a multi-class classification problem) with the different standard machine and deep learning methods. Finally, we assessed the predictive performance of the Cox proportional hazard and random survival forests methods based on a signature obtained using three gene mutations (KRAS, BRAF, and TP53). However, the most significant limitation lies in the sample size being small, and there is a lack of using independent data for validation. Also, we did not consider different features such as methylation and mutation data. Moreover, it is unfortunate that the study did not include detailed simulation studies to compare the traditional statistical and machine learning methods.

Overall, the most prominent finding to emerge from this investigation is that combining different data sources leads to more robust statistical significance. Also, the stacking approach is more reliable and promising compared to a single machine or deep learning. Furthermore, the RSF is a proper and striking method for survival analysis since it does not depend on any model assumptions.

Chapter 1

Background

Statistical, machine and deep learning methods have been applied in broad domains, including the medical field. These methods have a massive impact on healthcare by providing the support for decision making to the specialist in diagnosis and prognosis of patient status and disease progression (Kononenko, 2001; Magoulas & Prentza, 1999). Non-communicable diseases (NCDs) remain a challenge confronting the world in the 21^{st} century, especially in the developing countries where resources are limited (Organization et al., 2014). In a recent report by WHO, it is reported that 71% of total global deaths are caused by NCDs (Organization et al., 2018, 2020), with the very high socio-economic cost. Recent global public health studies exhibit an epidemiological paradigm transfer from infection to non-communicable diseases, which include cancer (Alwan et al., 2011). Cancer diseases are deemed the most destructive among all NCDs, and they are rated second to malaria as the main cause of death in developing nations (Jemal et al., 2011; Torre et al., 2015; Bray et al., 2018; Moten et al., 2014; Organization & for International Tobacco Control, 2008). In 2015, 8.8 million deaths were caused by cancer diseases globally. WHO estimated that by 2020 15 million new cases will be added to the cancer burden (Olsen, 2015; Morhason-Bello et al., 2013). Cancer diseases occur in many different types affecting all members of the community (men, women, children, and the elderly), where the general mechanism of cancer

disease etiology is uncontrolled cells proliferation that lead to a malignant or cancerous tumor such as breast, lung, and colorectal cancer, and abnormalities at the molecular level such as leukemia (Olsen, 2015; Morhason-Bello et al., 2013). There are several risk factors associated with cancer diseases, including tobacco, alcohol, obesity, and others (Halpin et al., 2010). However, earlier detection and accurate diagnosis of cancer symptoms increase the probability of curing the condition, and this has become the best strategy in fighting the disease.

In the past few years, a vast amount of cancer data have been generated through new high throughput microarray technology. Traditional clinical approaches were limited in data generation and lacked the capacity to handle such a massive scale of data. Therefore, computational methods have been introduced to biomedical investigations which include genes/biomarkers selection of cancer types and stages of the disease. Many computational and analysis tools have been developed based on different statistical and machine learning strategies and data science approaches. These tools have been applied in a broad range of biomedical investigations and have profoundly impacted disease diagnoses and prognoses. This has led to an increased capacity to select the best genes/biomarkers that accurately aid disease classification and prediction which in turn assist clinicians in choosing the suitable therapy strategies for cancer patients (Abusamra, 2013).

1.1 Cancer

Cancer is a disease that can happen in any part or organ in the human body where the cell growth is uncontrollable and can grow in different organs or parts of the body. However, cancer is among the most main causes of mortality and is also considered a significant disease burden worldwide (WHO, 2021a; Mohammed, 2018). The WHO predicted approximately 10 million deaths and 19.3 million new cancer cases in 2020. Therefore, in various countries, cancer is continuing to be a leading cause of death, and it exceeded the mortality caused by stroke and

coronary heart disease (Sung et al., 2021).

Scientists in the united states and worldwide put a lot of effort in the techniques that prevent, diagnosis, and treat cancer to improve survival of the patients. However, more research is needed for the success of immunotherapy and precision medicine. These researches use the techniques and tools that analyze the big data to come up with an accurate decision regarding the cancer disease (ACS, 2019).

Cancer detection relies on many methods, such as physical exams, laboratory tests, imaging tests, etc. A lot of research has been done on cancer classification. These researches used molecular-level investigation to provide an accurate and systematic diagnosis for various cancer types. Great insight regarding the cancer classification problem can be provided by genes data that determine their characteristics. The gene data include gene expression, which is used in cancer prediction, diagnosis, and drug discovery which are very important for cancer therapy. Also, this gene data can provide insight into the function of the genes and the interaction between them in a different situation by observing the behavior of gene-gene expression data under various conditions (Tarek et al., 2017).

1.2 High-dimensional Data

In high-dimensional data the number of the cases or the instances n is much less than the features or covariates p, that is, $n < p$. Most of the classical statistical inference research can be implemented when there is a small number of covariates "features" (Bühlmann & Van De Geer, 2011). With the recent explosion in data storage and computational tools, the high-dimensional data problem uprising has crucially occupied common statistical research (Ayesha et al., 2020). Recently, transcriptomics, epigenomics, and genomics studies have depended on high throughput sequencing (HTS) technologies. The HTS technologies involve parallel sequencing multiple DNA molecules, which enable the simultaneous sequencing of large amount of DNA molecules (Churko et al., 2013). Also, gene expression is controlled dynamically, where the differences in the transcription and translation

can cause a major functional changes within the cell. The mutation in the DNA sequence transcription and translation can compromise the function of the cell and cause various disease pathologies. Microarray and RNASeq are the most famous technologies that produce a massive amount of gene expression data.

1.3 Genomic Data

Genomic data grew exponentially over the last decade (Cerami et al., 2012; Chin et al., 2011b). It stands for the genome and the complete set of DNA information of an individual. This data stores the genomes of living things in the bioinformatics field (Mulder et al., 2017). A large amount of storage is required for the high-dimensional genomic data, and also purposely-built algorithms are needed to analyze them. The study of the genome helps to know the gene's interaction with each other and the environment. Moreover, it also allows us to understand the form of particular diseases, such as cancer, diabetes, and heart disease. Therefore, it is considered to be a new way to diagnose, treat, and prevent disease (Chin et al., 2011a).

1.4 Genetic Information

The human cells, excluding the red cells, have a nucleus. The nucleus has chromosomes that hold an individual's genetic information. The chromosomes arise from deoxyribonucleic acid (DNA) and proteins and contain hundreds to thousands of genes. These genes are organized in a specific sequence and location (Locus). The normal human cells nucleus carry 23 pairs of chromosomes, these pairs one chromosome from the mother, and one from the father makes a total of 46 chromosomes. Moreover, the unit that carries the genetic information is the DNA, whereas the protein components provide different functions (Wang, 2016; Datta & Nettleton, 2014; Ziegler et al., 2010). DNA is a group of molecules that carry and transmit all the essential genetic instructions to build and maintain an organism.

There are two strands in the DNA that have a linear backbone of alternating sugar (deoxyribose) and phosphate residues. The deoxyribose has five carbon atoms that are sequentially numbered from 1 to 5. The DNA sequence has four bases that are known as nitrogenous bases. These four bases are thymine (T) and cytosine (C) being pyrimidines, and adenine (A) and guanine (G) being purines. Each base with a unit of one sugar is called a nucleoside. Each nucleoside with a phosphate group tied to the carbon atom 5 or 3 makes one nucleotide (Wang, 2016; Ziegler et al., 2010). Moreover, the two strands of the DNA molecule are linked by a hydrogen bond between two opposing bases of the two strands. In addition to the DNA in chromosomes, another genetic information carrier is called ribonucleic acid (RNA) in the nucleus and the surrounding plasma of the cell. The RNA is created similarly to the DNA, with four main differences, which include that, it is contained only a single strand. In contrast, the single strand has the sugar component composed of ribose instead of deoxyribose. Lastly, RNA has uracil (U) as an alternative to thymine.

1.5 Transcriptome Analysis

The transcriptome can be defined as a set of messenger RNA (mRNA) molecules expressed by an organism under specific conditions (Lowe et al., 2017). Studying the transcriptome helps us interpret the functional and structural elements of the genome and comprehend human biology and diseases (Nguyen, 2020).

The whole transcriptome studies began in early 1990, consequently the technological advances made the transcriptomics a widespread in the biological science in the late 1990. The transcriptomics offers essential insights on gene structure, expression, and regulation and has been widely studied in the biological sciences. There are two crucial modern techniques for transcriptome analysis. These two techniques are DNA microarray and RNA sequencing technology. The DNA microarray quantifies a set of sequences that are predetermined. The RNA sequencing (RNASeq) technology capture all sequences through a high-throughput

sequencing (Lowe et al., 2017; Wang et al., 2019a). The RNASeq is considered to be the preferred and dominant transcriptomics technique in the recent years (Marco-Puche et al., 2019; Wang et al., 2009b).

Microarray has several benefits, these benefits include facilitating the analysis of a huge amount of genes from multiple samples and assess the incidence of a particular marker in tumors (Govindarajan et al., 2012). The technology that measures the DNA or uses it as a detection scheme is the microarray technology (gene chip, DNA chip, or biochip). Typically, a gene or a known DNA sequence is arranged in an order in the DNA microarray, based on the principle of hybridization between the nucleic acid strands. The microarray technology principle produces several copies of the mRNAs corresponding to a particular gene through transcription. After that, the mRNAs create the corresponding protein by a process called translation. The mRNA is then converted into cDNA form, whereas the cDNA is labelled by fluorochrome dyes Cy3 (green) and Cy5 (red). The unknown DNA molecules are cut into fragments by restriction endonucleases; and fluorescent markers are attached to these DNA fragments. These are then allowed to react with probes of the DNA chip. Then the target DNA fragments along with complementary sequences bind to the DNA probes. The remaining DNA fragments are washed away. After that, a comparison of the two fluorescence intensities can be made to identify the genes that are differentially expressed between two samples (Jaluria et al., 2007). Also see Figure 1.1 as shown by Sagar Aryal (Aryal, 2018). The DNA microarray technology has two different types, these types are cDNA-based microarrays and oligonucleotide based microarrays.

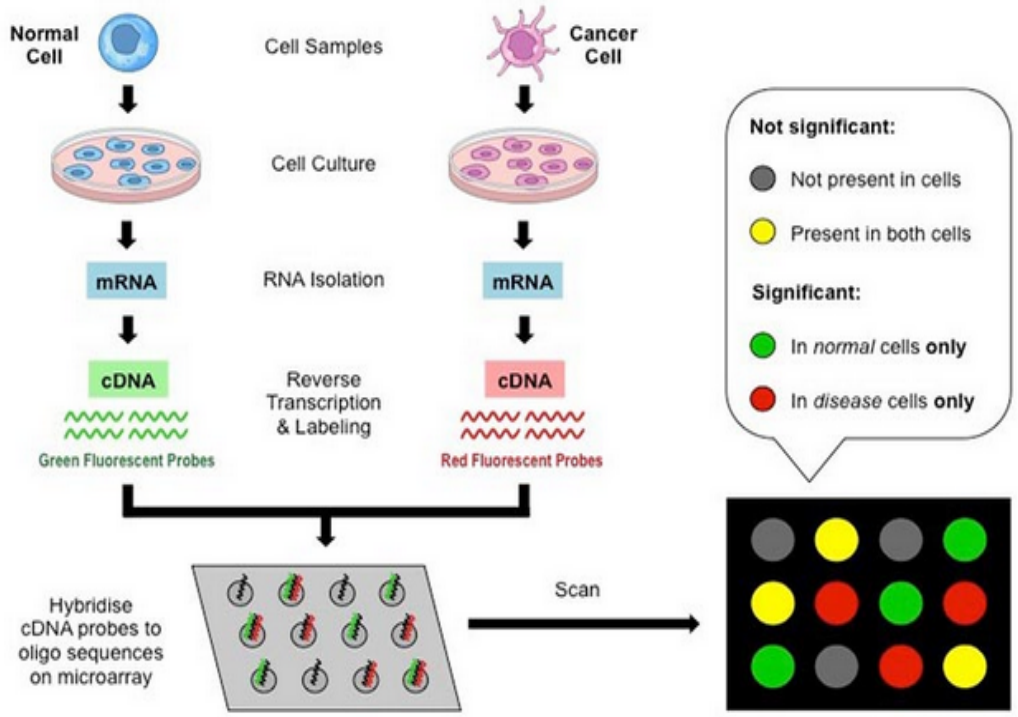


Figure 1.1: Microarray analysis steps [Image Source: BioNinja]

The RNASeq technique utilizes next-generation sequencing (NGS) to examine the entire transcriptome (Wong, 2017; Kulski, 2016; Wang, 2016; Harbers & Kahl, 2011; Kwon & Ricke, 2011). RNASeq provides an accurate measurement of level of transcripts and it is a high- throughput alternative to the traditional RNA/cDNA cloning and sequencing approaches (Wang et al., 2009b). The RNASeq technology procedure steps are shown by the *RNA-Seq Bolg* (`https://www.rna-seqblog.com/introduction-to-rna-sequencing-and-analysis/experiment/`) in Figure 1.2.

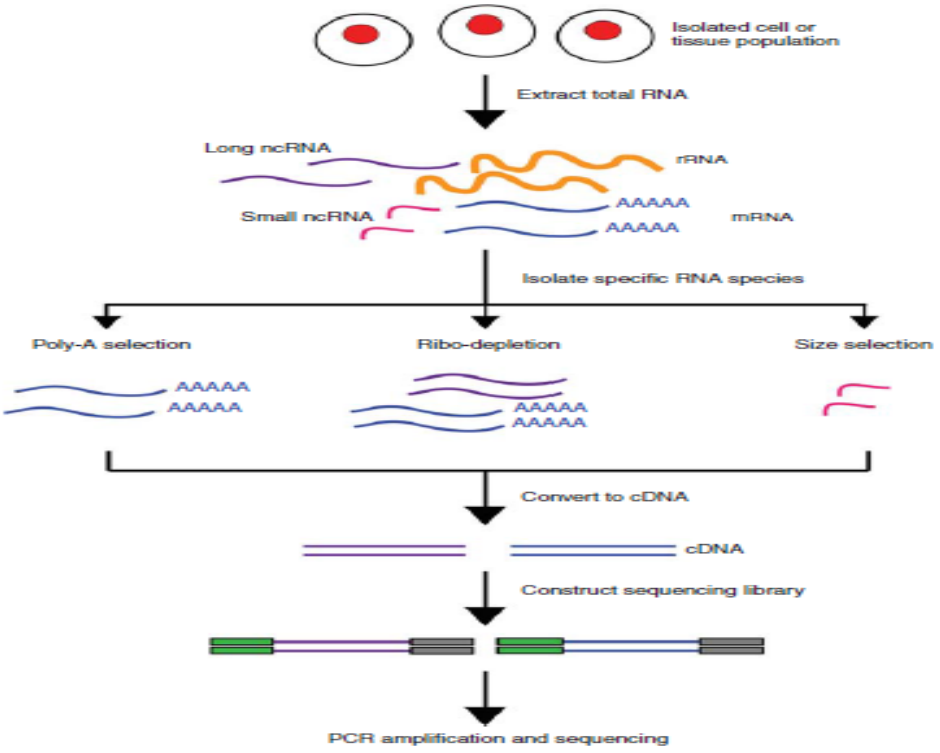


Figure 1.2: The RNASeq technology workflow steps [Image Source: RNASeq]

More recently, literature show that RNASeq data is better than microarray data in terms of high quality, accurate estimates, producing millions of sequencing reads, but it should be acknowledged that the two types of data follow different methodologies, referenced from Castillo et al. (2019); Rai et al. (2018); Zhang et al. (2015). Even though RNASeq techniques are expensive, they are better than microarrays because they allow the detection of single nucleotide variation, and they do not require knowledge of genomic sequence. Also, they provide isoform-level expression measurements, quantitative expression levels, and other broader dynamic ranges than microarrays, referenced from Castillo et al. (2019). Moreover, RNA-Seq allows the identification of novel transcripts, reduces the background signal, and enhances specificity and sensitivity as reported in Dong et al. (2016).

1.6 The Transcriptomic Data Notations

The evolution of transcriptome technologies allows to massively produce thousand of genes or transcripts expression levels simultaneously with meager cost (Lowe et al., 2017). However, understating and interpreting the results of this massive data is needed and important. Here, we present the notation and the shape of the transcriptome data.

The gene expression data is stored into a matrix G with dimension $p \times n$, where the p represents the number of genes/covariates (variables or features) and n is the number of the samples/patients or tissue (observation). The indexes of the genes and samples are symbolized by i and j, respectively. However, g_{ij} embodies the expression level for gene i in sample j ($i = 1, 2, 3, \ldots, p$ and $j = 1, 2, 3, \ldots, n$). The matrix shape of the gene expression is shown in Figure 1.3 below

$$\begin{pmatrix} & SAMANN_1 & SAMANN_2 & SAMANN_3 & \ldots & SAMANN_n \\ GANN_1 & g_{11} & g_{12} & g_{13} & \cdots & g_{1n} \\ GANN_1 & g_{21} & g_{22} & g_{23} & \cdots & g_{2n} \\ GANN_1 & g_{31} & g_{32} & g_{33} & \cdots & g_{3n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ GANN_1 & g_{p1} & g_{p2} & g_{p3} & \cdots & g_{pn} \end{pmatrix}$$

Figure 1.3: The gene expression matrix. GANN: Gene Annotation, SAMANN: Sample Annotation.

1.7 Differential Gene Expression (DGE) Analysis

The differential gene expression is the process of defining the genes that show differences in the expression levels between specific conditions (Crow et al., 2019). Usually, we use statistical testing to determine whether an observed difference in the expression levels of a gene is significant. There are many statistical methods

such as t-test (Kim, 2015), limma (Smyth, 2004), edgeR (Robinson et al., 2010) and DESeq based on negative binomial distributions (NBD) (Anders & Huber, 2010), among others.

1.7.1 Hypothesis Testing Procedure

The statistical inference aims to draw conclusions about a population using data collected from a sample of that population. However, hypothesis testing is a way to evaluate the strength of evidence found in a sample for making general conclusions about the population (Banerjee et al., 2009; Browner et al., 2001). Testing hypotheses begins with stating the research question into null and alternative hypotheses, H_0 and (H_1 or H_a), respectively (Paiva, 2010; Browner et al., 2001).
Consider the case where we are looking for differentially expressed genes between two or more different tumor types (or conditions). Suppose that μ_i^k is the mean of the expression levels in condition k in gene i, where $k = 1, 2, \ldots, c$ and $i = 1, 2, \ldots, p$. These hypotheses in the differential gene expression analysis can be stated as:

$$H_0 : \text{The mean expression levels between the conditions are equal}$$

$$\text{versus}$$

$$H_a : \text{At least two mean expression levels are not equal}$$

that is

$$H_0 : \mu_i^1 = \mu_i^2 = \ldots = \mu_i^c$$

$$\text{versus}$$

$$H_a : \text{At least } \mu_i^k \neq \mu_i^l, \text{ for } k \neq l$$

Thereafter, we set the significance level α, which represents the probability of rejecting the H_0 when H_0 in fact true (Massey & Miller, 2006). Subsequently, we fit the model in order to estimate the parameters that are associated with g_i for each condition. Finally, we compute the test statistic values for each gene g_i and their corresponding p-values, which we use to compare with the significant level α in order to determine whether to reject or fail to reject H_0 (Massey & Miller, 2006). In

the case that the p-value is less than α we reject the null hypothesis H_0, and we conclude that there are statistical differences in the mean expression levels between the conditions in gene g_i, and this conclusion is said to be statistically significant. There are two types of errors that can occur during the hypothesis testing process (Paiva, 2010). Thus, there is a chance to make mistakes; however, the possible outcomes are given in Table 1.1 below:

Table 1.1: Four outcomes for making a decision. The decision can be either correct (correctly reject or fail to reject null) or wrong (incorrectly reject or fail to reject null)

	Actual Situation (Truth)	
Test Statistic (Decision)	H_0 **is True**	H_0 **is False**
Fail to reject H_0	Correct Decision $(1-\alpha)$	Type II error (β)
Reject H_0	Type I error (α)	Correct Decision $(1-\beta)$

1.7.1.1 Type I Error

Type I error is also known as "false positive", and it is defined as the error when a null hypothesis is rejected while it is actually true.

1.7.1.2 Type II Error

Type II error is also known as a "false negative", is the error of failing to reject a null hypothesis while it is actually false.

Moreover, to explain the Type I and Type II error, let us consider the following example in our case (the gene expression data), the mean expression levels of gene g_i across the conditions are equals (not differentially expressed) but the test statistic results show that this gene g_i is differentially expressed (the mean expression levels of the conditions are not equals), this is the case of Type I error. On the other other hand given the gene expression levels for gene g_i are differentially expressed between the conditions but the test statistic results show that the p-value is larger than α we have a case of Type II error. Although Type I and Type II errors cannot be

completely prevented, test statistics use the significance level α to control the likelihood of creating Type I errors.

1.7.1.3 Multiple Testing

Performing a single statistical test for each gene has numerous drawbacks. The most notable is that it involves a considerable number of hypothesis tests, which might result in a high number of erroneously significant findings. Since we are dealing with genomics data containing lots of genes (features), sometimes tens to hundreds of thousands, this leads to lots of hypothesis tests. For instance, in general, when performing N hypothesis tests, the probability of finding at least one Type I error (false positive) is calculated by equation (1.1) (Herzog et al., 2019):

$$\begin{aligned} &P(\text{finding at least 1 error in the } N \text{ tests}) \\ &\qquad = 1 - P(\text{Not finding an error in the } N \text{ tests}) \\ &\qquad = 1 - (1-\alpha)^N. \end{aligned} \tag{1.1}$$

Therefore, the proportion of finding at least one error in the N hypothesis test is around $1-(1-\alpha)^N$ (also known as family-wise error rate or (FWER)) (Pan, 2013), this more likely produces false positive when the number of hypothesis tests increases (Herzog et al., 2019). Hence, that would be a major issue in the genomics studies, which require tens to hundreds of thousands of hypothesis tests.
There are various methods, such as Bonferroni adjustment and Benjamini-Hochberg adjustment, among others that deal with the multiple testing problem to adjust the α in the genomics studies in order to ensure that the probability of finding at least one significant findings by chance is still smaller than the significance level α.

Bonferroni Adjustment: it is also known as Bonferroni correction, which controls

the FWER in the case of the multiple hypothesis testing (Herzog et al., 2019; Vickerstaff et al., 2019; Pan, 2013; Bland & Altman, 1995). This method is one-step procedure, which implies that for each test the p-value must equal to its α divided by the number of tests performed as given in equation (1.2) below:

$$\text{Adjusted } \alpha = \frac{\alpha}{\text{number of hypothesis tests}}. \tag{1.2}$$

Benjamini-Hochberg (BH) Adjustment: This test controls the false discovery rate (FDR) (Benjamini & Yekutieli, 2001; Benjamini & Hochberg, 1995). The BH procedure will decrease the number of false positives as follow (Benjamini & Yekutieli, 2001):

1. Calculate the p-values corresponding to all hypothesis tests and put them in ascending order.
2. Assign ranks to the p-values, starting from 1 for the smallest p-value, 2 for the second smallest p-value, etc.
3. Calculate the Benjamini-Hochberg critical value using the equation $\frac{i}{m} \times Q$, where i is the rank of the p-value, m is the total number of hypothesis tests, and Q is the percentage of the desired false discovery rate.
4. Compare the p-values to the critical BH from step 3.
5. Find the largest p-value that is smaller than the critical value.
6. Finally, the p-values that are smaller than this p-value in step 5, are considered significant.

1.7.2 Differential Gene Expression Data Modeling

An important step of the statistical test in the genomics data is choosing the probabilistic model. RNASeq and Microarray technologies produce different forms of gene expression data. The data that is created using RNASeq is discrete, while

the data that is produced using microarray is continuous as explained in Zararsız et al. (2017). What is typical between the two technologies is that both NGS and microarray technologies usually produce big datasets that have small number of cases (small sample size), and each case has a large number of genes. Differential expression analysis has been used to find the most relevant genes that highly distinguish between two or more conditions using data produced by these technologies. Hence, for the microarray, several methods have been developed for this purpose, including *limma* (Smyth, 2004), which is based on the normal distribution. Unfortunately, due to the type of data produced by the RNASeq technology, this method is not appropriate for the RNASeq data. In general, the microarray is commonly assumed to follow the normal distribution. However, RNASeq data is modeled using Poisson and NBD (Wang et al., 2010; Di et al., 2011; Auer & Doerge, 2011), which are considered as the most suitable for modeling count type data.

The Poisson distribution does not consider the biological variation in the data due to it is assumptions which assume that the mean and the variance are equals (Robinson & Smyth, 2007). However, this is not the case in the RNASeq data, where the low expressed genes have higher variance than high expressed genes. Hence, to account for this issue, the NBD is used to replace Poisson for modeling this kind of data. The NBD is a discrete probability distribution that models the number of successes in a sequence of independent and identically distributed Bernoulli trials before a specified number of failures. In addition, it has two parameters, the mean and variance. The RNA-seq data can also be transformed using a simple logarithm, the regularized logarithm (Love et al., 2014), or a more complex variance stabilizing transformation (Anders & Huber, 2010). There are several transformation methods including *voom* and *limma-trend* that allow RNASeq to use many methods developed for microarray that are not available for RNA-seq (Law et al., 2014).

Many statistical software have been used for the differentially expressed genes detection based on the distributional assumption of RNA-seq count data, including

DEGseq, *DEGseq2*, *edgeR*, and *limma*. Before performing statistical analysis, it is essential to consider normalization, which has a huge impact on the differential expression analysis result (Dillies et al., 2013; Bullard et al., 2010). Hence, it is designed to identify and correct the presented technical biases resulting from the library preparation protocols and sequencing platforms.

1.8 Statistical Learning

Statistical learning plays a crucial role in many areas of science, finance, industry, medical among others (Vapnik, 2013; Friedman et al., 2001). It encompasses a diverse range of methods for understanding data. These methods can be either supervised or unsupervised. The most common approach is supervised learning which aims to build a model that predicts or estimates output based on given inputs (Liu, 2011; Cunningham et al., 2008). In contrast, in the unsupervised learning, there are inputs but no given outputs; the goal is to learn the pattern and structure from data (James et al., 2013a).

1.8.1 Supervised Learning

A supervised learning approach is a method that learns a function mapping (f) of inputs variables G to particular outputs variable Y, using labelled observations which are commonly known as *training set* (Cunningham et al., 2008). Thereafter, it uses this mapping function to predict the outputs for new data, which is called *testing set*. Consequently, the idea behind supervised learning is to use a learner (method) to learn from historical data (*training set*) to identify the new observations in the testing set with the highest possible accuracy. However, the goal of classifiers (learners) is to define rules from the existing historical data and then use them to classify and predict unseen data (Learned-Miller, 2014; James et al., 2013a; Friedman et al., 2001). Consider that the *training set* contains n samples pairs $(G_1, y_1), (G_2, y_2), \ldots, (G_n, y_n)$, where the G_i is the set of gene expression levels measurements of a single observation data point in our case, and y_i is the class label

which is a cancer type, sub-types, or normal vs tumor samples among others in our case. In contrast, the *test set* is another data with m samples with unknown class labels $(G_{n+1}, y_{n+1}), (G_{n+2}, y_{n+2}), \ldots, (G_{n+m}, y_{n+m})$, in this case the class $y_{n+1}, y_{n+2}, \ldots, y_{n+m}$ are unknown. As mentioned previously, the aim of the supervised learning is to perform correct classifications of the outcomes for the *testing set* based on the trained model from the *training set*. The supervised learning has two tasks the *classification* when the output type is categorical, and the *regression* when the output type is continuous (Friedman et al., 2001). The popular approach for calculating the accuracy of estimated function $(\hat{f})$ is the training error rate as given below (James et al., 2013a):

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i), \tag{1.3}$$

where $\hat{y}_i$ is the predicted class label for the i^{th} observation using $\hat{f}$, while I is an indicator variable that equals 1 if $y_i \neq \hat{y}_i$ and 0 if $y_i = \hat{y}_i$, therefore this equation calculates the fraction of incorrect classifications. Moreover, the test error rate can be calculated by (James et al., 2013a):

$$\frac{1}{m}\sum_{i=1}^{m} I(y_{n+i} \neq \hat{y}_{n+i}), \tag{1.4}$$

where $\hat{y}_{n+i}$ is the predicted class label that results from applying the classifier to the test observation.

There are different types of supervised models, including k-nearest neighbor (kNN), logistic regression, naive Bayes (NB), support vector machines (SVM), neural networks (NNs), linear discriminant analysis (LDA), Poisson linear discriminant analysis (PLDA), and negative binomial linear discriminant analysis (NBLDA) among others.

1.8.2 Unsupervised Learning

Unsupervised learning depicts the situation for every observation in the data that does not contain related label information. The aim here is to directly infer the labels (James et al., 2013a; Friedman et al., 2001). Unsupervised learning is implemented in various ways, including data reduction, compression, visualization, density estimation, clustering, and preprocessing (Biehl, 2019). There are various types of unsupervised models, these models include independent component analysis (ICA), singular value decomposition (SVD), principal component analysis (PCA), K-means clustering, and hierarchical clustering among others.

1.9 Machine Learning (ML)

Machine learning is a subgroup of artificial intelligence (AI) algorithms that are designed to simulate the way that humans learn and getting experience to become more accurate at predicting outcomes (Zhang, 2020; El Naqa & Murphy, 2015; Wang et al., 2009a). ML and statistical learning methods are very closely associated fields. Machine Learning is considered as a method of statistical learning, and also allows for supervised and unsupervised learning. ML seeks to investigate computer-assisted self-improvement strategies for acquiring new information and abilities to identify current knowledge and constantly to improve performance and achievement (Wang et al., 2009a). Moreover, Machine learning algorithms are based on mathematical models and are associated with computational statistics, which primarily emphasizes making predictions using computers. Nowadays, ML is capable of analyzing large amounts of complex data and identifying patterns as a result of technological developments. It creates a model from training data and allows researchers to make predictions for new or unknown data using the same model. ML has also made significant contributions to bioinformatics, cancer detection, medication development, traffic pattern analysis, pattern recognition,

computer vision, spacecraft engineering, finance, entertainment, and computational biology (biomedical and medical applications), among others (Leung et al., 2015; El Naqa & Murphy, 2015; Wang et al., 2009a). However, it is hard to train ML model as data becomes more complicated and extensive, yet the improvement in the machine learning algorithms has allowed many issues to be solved.

1.10 Classification

The practice of classifying or separating data instances into distinct groups is known as classification. A classification is a form of supervised learning in which data components are assigned to a certain class. It is the process of predicting the class of a given set of data, with the classes being referred to as targets, labels, or categories. Supervised machine learning includes several different types of classification tasks, such as binary and multi-class classification. The binary classification aims to predict one of two outcomes (classes), while the multi-class classification involves predicting one of more than two outcomes (classes). In general, a n-labels classifier may classify n possible outcomes. However, in some cases, the binary classifier can be used as a multi-class classifier (this is called as *one versus all (OVA) or one versus rest (OVR)* approach. The idea behind this approach is to build a model for each class against the rest; thus, the number of obtained models is equivalent to the number of the classes, and then select the model that gives the highest probability and assigns that class to the given sample. In this thesis, we deal with binary and multi-class classification problems.

1.11 Deep Learning (DL)

Deep learning is a novel machine learning approach with outstanding performance in unstructured data compared to traditional machine learning methods. It has algorithms and models that can discover characteristics from data at various levels in a step-by-step manner (Mathew et al., 2020). The growth in data accessibility and

the improvement in the hardware power and storage capacity make deep learning even more popular, especially the transfer learning which uses pre-trained neural networks that are trained on a massive amount of data (Bengio et al., 2017). Neural networks are algorithms meant to identify patterns and are roughly modeled to emulate how the human brain works (Hurwitz & Kirsch, 2018). A neural network contains three or more layers including, an input layer, one or many hidden layers, and an output layer.

The DL is similar to a traditional neural network except that it has many more hidden layers. Its hidden layers will increase as the problem becomes more complicated (Hurwitz & Kirsch, 2018). The DL has been applied widely in many areas such as automatic speech recognition, image recognition, natural language processing, computer vision, drug discovery and toxicology, customer relationship management, recommendation systems, high-dimensional time series data, and bioinformatics (Heaton, 2020; Sk et al., 2017; Hordri et al., 2016; Najafabadi et al., 2015; Deng & Yu, 2014). The differences between artificial intelligence, machine learning, and deep learning are shown in Figure 1.4.

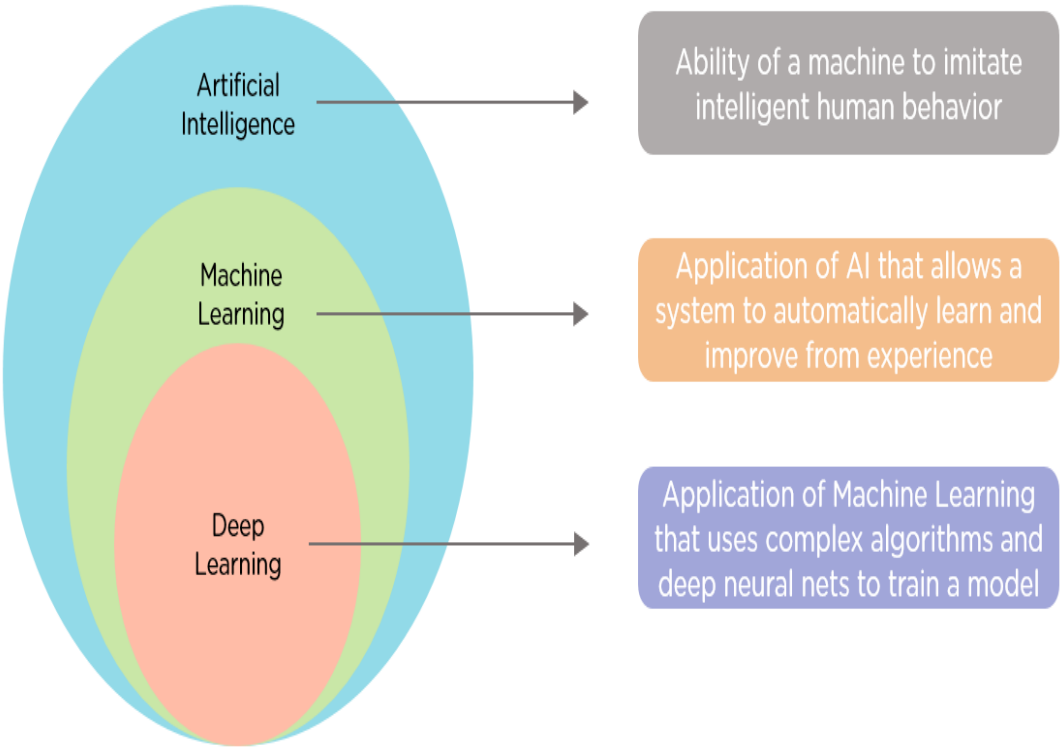


Figure 1.4: The difference between AI, ML, and DL [Image Source: Differences of AI, ML, and DL]

1.12 Re-sampling Methods

Resampling approaches are a collection of methods for repeating, drawing different samples from a training set, and refitting a model of concern on a test set to get more information such as accuracy, sensitivity, specificity, among others about the fitted model (James et al., 2013b). However, in current statistics and machine learning, resampling approaches are an essential tool. There are two primary resampling techniques, namely validation set, *K*-fold cross-validation (CV), and Bootstrap.

1.12.1 Validation Set

In general, the idea behind the validation set method is that the data is divided into training and validation (hold-out) sets. After that, we fit the model of concern

using the training set, and then the trained model is used to predict the outcomes of the sample in the validation set. The validation set technique is attractive due its simplicity and ease of implementation but it has two significant drawbacks in terms of the quality and amount of data seen in the training and validation sets. The first disadvantage is that the error rate is determined by which observations are included in the training and validation sets. Secondly, the fitted model is trained on a subset of the data (training set), and the model performs poorly on the training set with fewer samples. It also may suffer the problem of the error rate being overestimated or greater than that obtained when the model is fitted to the entire dataset. For these reasons the approach has been extended to counter some of these shortcomings.

1.12.2 *K*-Fold Cross-Validation (CV)

The CV is typically used for model assessment by estimating the error rate of a particular model, i.e., CV helps examine the capacity of prediction models to generalize and avoid over-fitting (Friedman et al., 2001; Hart et al., 2000). In addition, it is considered as one of the most commonly used data resampling approaches for estimating the actual prediction error of models and tuning model parameters (Berrar, 2019). In *K*-fold cross-validation we randomly divide the data into approximately *K* equal-sized subsets (Friedman et al., 2001). Thus, we fit the model using a part of the available data and use a different subset to validate that model's performance. In other words, the $K-1$ parts are used to train the model, and the last part is used as a test set. In practice, the most common number of folds is 5 or 10 depending on the amount of the available data (Friedman et al., 2001). This procedure is iterated until each of the *K*-folds act as a test set. Finally, the error rate is calculated by averaging the estimation obtained from each of the *K*-folds. Consider $k : \{1, \ldots, N\} \mapsto \{1, \ldots, K\}$ be the indexing function that identifies the subset in which an observation i is assigned randomly. The $\hat{f}^{-k}(x)$ is the fitted function that is computed using the k^{th} subset of the data removed that is considered as the test set. Therefore, the cross-validation estimate of prediction

error is given by:

$$CV(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-k(i)}(x_i))\,, \tag{1.5}$$

where L is the loss function.

When the $K = N$ then it is called *leave-one-out* cross-validation. In this case, the model is fitted using the whole data except the i^{th} case where $i = 1, 2, 3, \ldots, N$. However, the process is repeated N times until each case acts as a test set.

Overall, when there are a group of models indexed by a tuning parameter α, the K-fold cross-validation can be used to estimate the $CV(\hat{f}, \alpha)$ for each model by

$$CV(\hat{f}, \alpha) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-k(i)}(x_i, \alpha)). \tag{1.6}$$

The equation (1.6) above, gives an estimate of the test error curve, then we choose $\alpha = \hat{\alpha}$ that minimizes the test error curve. Finally, we fit the selected model $f(x, \hat{\alpha})$ to the entire data.

In the classification problem with a large number of variables such as genomics data, Friedman et al. (Friedman et al., 2001) shows the correct strategy of doing K-fold cross-validation as follows:

1. Divide the samples into K cross-validation folds (groups) at random.
2. For each fold $k = 1, 2, \ldots, K$
   (a) Find a subset of "good" predictors that show fairly strong (univariate) correlation with the class labels, using all samples except those in fold k.
   (b) Using just this subset of predictors, build a multivariate classifier, using all samples except those in fold k.
   (c) Use the classifier to predict the class labels for the samples in fold k.

1.12.3 Bootstrap

A bootstrap method is a resampling approach that uses sampling with replacement for statistical inference (Dixon, 2006). Assume we want to fit a model on a training set of n samples pairs as indicated before $(G_1, y_1), (G_2, y_2), \ldots, (G_n, y_n)$. Thus, the idea behind the bootstrap approach is to draw a sample with replacement randomly. Each bootstrap has the same size as the original sample of size n and this is repeated B times (B bootstrap datasets). Each of these bootstrap samples is used as the training set (Chernick et al., 2011; Friedman et al., 2001). Then, use these B bootstrap datasets to refit the model and measure its performance over the B replications. The prediction error is estimated by

$$\hat{Err}_{boot} = \frac{1}{B}\frac{1}{N}\sum_{b=1}^{B}\sum_{i=1}^{N} L(y_i, \hat{f}^{*b}(x_i)). \tag{1.7}$$

1.13 The Curse of Dimensionality

The curse of dimensionality was initially introduced by Richard Bellman (Kuo & Sloan, 2005; Bellman, 1966). Unfortunately, in real-life data such as genomics data, the number of the variables could be exceedingly high in terms of multiples of thousands or more, while the number of samples is only a hundred. Thus, in this situation, the curse of dimensionality becomes real and leads to an exponential increase in computational effort, large waste of space, and poor visualization capabilities (Venkat, 2018). The curse of dimensionality states that the error rises as the number of variables increases. Thus, most of the methods are harder to design. However, several techniques have been invented to reduce the effects of high-dimensional data, and these methods are known as features selection or reduction.

1.14 Variable (Feature) Selection

Feature selection is a preprocessing method for identifying a subset of variables (covariates) that improves comprehension and prediction performance (Li et al., 2017; Jović et al., 2015). The feature selection methods have been proved to be valuable and efficient for data preparation by choosing the most informative characteristics and removing unnecessary and redundant features, particularly for high-dimensional data (Li et al., 2017; Guyon et al., 2008; Saeys et al., 2007; Guyon & Elisseeff, 2003; Kira & Rendell, 1992). At the same time, feature selection is being used for both supervised and unsupervised learning. In this work, we will focus on supervised learning (classification), where the class labels are known. However, the feature selection aims to get a better understanding of the underlying processes that created the data. Therefore, it can enhance model performance and prevent overfitting problems. Also, it can help in developing more efficient and cost-effective models (Saeys et al., 2007).
There are three standard types of feature selection approaches based on how a combination is made between the feature selection methods and the classification models. These types include filters, wrappers, and embedded approaches.

1.14.1 Filter Approach

The filter approach evaluates the relevance of the variables based on the intrinsic characteristics of data using statistical measures. However, the selection of the features is independent of the classification model that is utilized (Guyon et al., 2008; Sánchez-Maroño et al., 2007). Typically, it calculates a score for all the variables and then ranks them (Lazar et al., 2012). The filter approach can select predictors by either choosing predictors that pass a given threshold or keeping the number of features that one desires. The filter methods contains parametric methods such as *t-test*, *chi-square test* among others, or non-parametric methods including *Wilcoxon sum-rank statistics* among others. The most notable benefit of filter methods is that their calculation is quick and straightforward, making them

ideal for use as an initial processing step in a high-dimensional dataset.

1.14.2 Wrappers Approach

The wrappers methods select the features based on the classification model performance. They aim to find the most optimal features that give the best possible performance with a particular learning algorithm (Guyon et al., 2008; Kohavi & John, 1997). However, the wrappers methods search iteratively uses a machine learning method for a subset of features and then evaluate them based on the model performance measures such as accuracy, sensitivity, specificity, ..., etc (Mlambo et al., 2016; Lazar et al., 2012). These approaches are known as greedy algorithms because they seek to discover the best possible combination of characteristics that result in the best possible model performance; thus, they are computationally costly. Moreover, they have a high possibility of overfitting since they contain training of machine learning models with different combinations of features. Wrapper techniques are typically more accurate compared to the filter methods (Mlambo et al., 2016; Inza et al., 2004). Wrappers methods include Recursive Feature Elimination (RFE), Genetic Algorithm (GA) among others.

1.14.3 Embedded Approach

The embedded methods are selection methods integrated as a part of the learning model (Jović et al., 2015; Guyon et al., 2008). These methods select the subset of features during the building of the model in order to reduce the computational time. Embedded methods combine both filter and wrapper methods (AlNuaimi et al., 2020). The most Common embedded methods are the tree algorithms such as Random Forest, ExtraTree, and regressions or classification combined with LASSO regularization.

1.15 Ensemble Approach

Ensemble methods are statistical and computational learning approaches that combine several classifiers typically by weighted or unweighted voting of their predictions to obtain more reliable and more accurate predictions in supervised and unsupervised learning problems (Way et al., 2012; Dietterich, 2000). Ensemble methods have been widely used in computational biology due to the unique characteristics in handling high-dimensional and complex data structures (Yang et al., 2010). Moreover, the ensemble learning has become a prominent machine learning technique in the last 20 years (Zhou, 2019). It has been an active area of research for constructing a good combination of models in supervised learning. However, the most critical finding is that ensembles are frequently more accurate than the base learners that comprise them.

Many methods for constructing ensembles have been developed. Ensemble methods are categorized into Sequential, Parallel, homogeneous, and heterogeneous. The most popular methods include boosting, bagging which part of the homogeneous ensemble, and stacking which falls in the heterogeneous ensemble methods. Ensemble techniques are particularly well suited to regression and classification, where they minimize bias and variance while increasing model accuracy (Dong et al., 2020; Sagi & Rokach, 2018).

1.15.1 Stacking Ensemble

Stacking (also known as stacked generalizations) is a type of ensemble learning where a varied group of models (heterogeneous models) are used as base learners and their prediction is combined by a meta classifier on the top layer (Chatzimparmpas et al., 2020; Sagi & Rokach, 2018; Wolpert, 1992). Noteworthy, stacking minimize the bias and the generalization error compared to single classifiers by combining different heterogeneous classifiers (Chatzimparmpas et al., 2020). It's structure is shown in Figure 1.5.

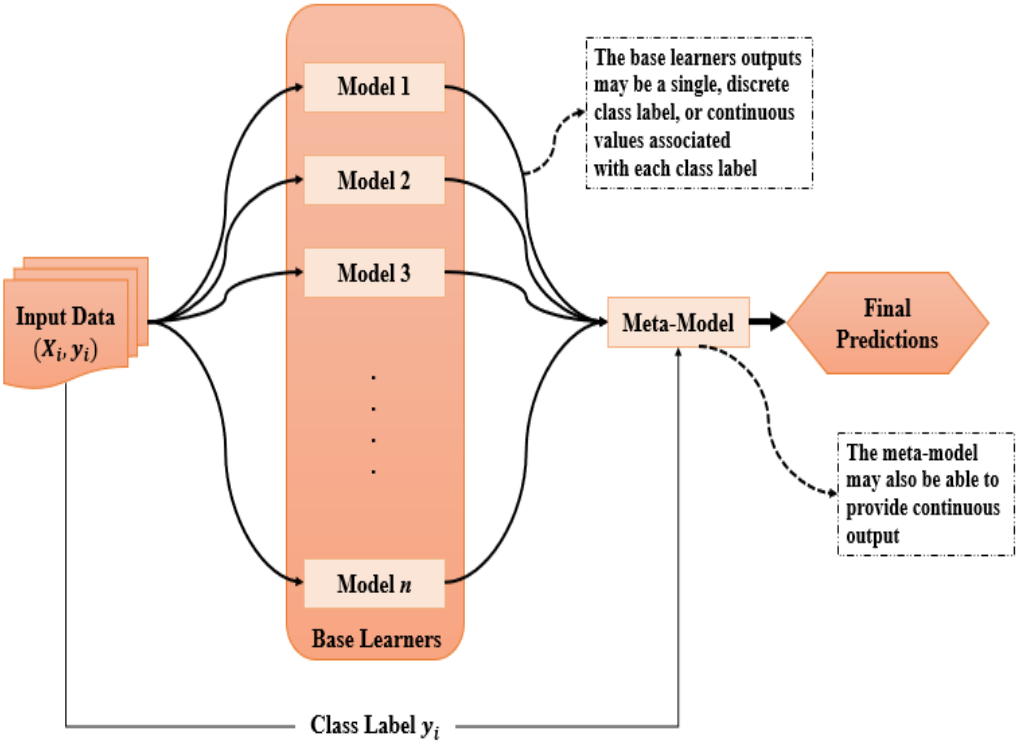


Figure 1.5: The general architecture for stacked generalization model.

The outputs obtained by the set of n base learners, combined with the labels of the input data, are utilized to train the meta-model that can anticipate the final predictions for unseen input. The meta-model seeks to learn from the knowledge that is obtained by the base learners, and it may be able to understand and exploit patterns and regularities in their output.

1.16 Imbalanced Data Problem

An imbalanced classification problem is defined as a classification problem where the distribution of instances across different classes is biased or skewed. The class with the fewer instances are known as the minority class, while the class that contains the higher number of instances is called majority class. In large application domains, such as medical diagnostics, the minority classes with fewer instances generally contain the critical data (Zhao et al., 2021). However, most of the traditional classification methods fail to deal with imbalanced data due to their

assumption of an equal distribution of classes, thus, it results in models that have poor predictive performance, specifically for the minority class (Kotsiantis et al., 2006). Many methods have been proposed for the class-imbalance problem including many different forms of resampling techniques such as random over-sampling and random under-sampling.

1.16.1 Over-sampling Technique

Over-sampling also known as up-sampling, is the method that tend to balance the classes by adding observations to the minority class in order to reduce the skew in the class distribution (Hoens & Chawla, 2013; Chawla, 2009; Kotsiantis et al., 2006; Chawla et al., 2002).

1.16.2 Under-sampling Technique

Under-sampling also known as down-sampling, aims to balance the class imbalance through eliminating and removing observation that belong to the majority class (Hoens & Chawla, 2013; Chawla, 2009; Kotsiantis et al., 2006; Chawla et al., 2002).
Overall, here we employed synthetic minority oversampling technique (SMOTE) and down-sampling methods for handling class imbalanced data problem.

1.17 Survival Analysis

Generally, we use logistic regression to study the association of the risk factors with presence or absence of a disease. However, sometimes we might be interested to know how the risk factors affect time to disease or any other event of interest. Thus, in this case the logistics regression fail. Survival analysis is an alternative technique used to analyze data in which the time until the event is of interest.

Censored Data: In survival analysis is an approach for analyzing data when the variable of interest is time until an event occurs. The biggest challenge in survival

analysis is that some observations will not experience the event by the end of follow up such due to end of study, loss to follow-up or withdrawal from the study. Therefore, we do not know the exact event time; this is known as censored data (or called censoring). In this case, our variable of interest is $y_i = \{t_i, \delta_i\}$, where t_i refers to as failure time, survival time, time-to-event, or follow-up-time, while δ_i shows whether the event is observed or not, 1 indicates that the event has been observed and 0 indicates that the observation has been censored. There are three main types of censoring, right, left, and interval. The right-censored time can be defined as the observation that where the observed time is less than the true event time. In contrast, the left-censored time is defined as the observations where the event was experienced before enrolment. On the other hand, when a random variable is not observed exactly i.e., lie within an interval in this case it is called interval censoring.

Survival and Hazard Functions: Survival and hazard functions play prominent roles in survival analysis. The survival function $S(t)$ is the probability that a patient survives longer than time t. While the hazard function $h(t)$ is the rate of failure (risk) at time t, given survival up to time t. Several techniques have been used for estimating survival function or survival curve such as Kaplan-Meier (KM) method (Ranstam & Cook, 2017).

1.18 Missing Data

Missing data (or missing values) normally occurs in the gene expression obtained through high-throughput sequencing technologies, therefore, that may lead to inaccurate findings (Farswan et al., 2020). Missing values can be defined as values of a variable of interest that are not observed (Kang, 2013). The impact missing data on the research can be severe and affect the statistical conclusions that can be drawn from data. The missing data result in several challenges including, reduction statistical power, cause bias in the estimation of parameters, reduction in the representation of the samples, and may complicate the statistical analysis in the

studies. Thus, missing values can lead to invalid conclusion (Dong & Peng, 2013). The missing data can occurs in different mechanisms, these mechanisms are missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR) (Rubin, 1976). Rubin (Rubin, 1976) developed the theory of missing data, each data point has some likelihood of being missing. The process that controls these probabilities is called the missing data mechanism or response mechanism. The model for the process is called the missing data model or response model. In the case all data points having same probability of being missing, then the data are said to be missing completely at random (MCAR). While, in the case that the data points having same probability of being missing only within groups identified by the observed data, then the data are missing at random (MAR). Consequently, the case of missing not at random (MNAR), occurs when the probability of being missing is neither MCAR nor MAR holds.

1.19 Imputation Techniques

Various missing data handling approaches are used to handle missing data such as complete case analysis (listwise deletion), available case analysis (pairwise deletion), mean imputation, single imputation, stochastic imputation, and multiple imputation (MI) among others. MI is widely used to handle missing data (Huque et al., 2018; Molenberghs & Kenward, 2007). MI is typically an iterative form of stochastic imputation that provides uncertainty about the missing data by generating various plausible imputed datasets and properly combining results obtained using the Rubin's rules for inference, where the standard errors used accounts for the variation within and between imputations (Carpenter & Kenward, 2012; White et al., 2011).

1.20 Evaluation Metrics

The methods performance evaluation is crucial in the statistical, machine, and deep learning fields to know the best model that fits the data and has achieved better performance. There are several metrics such as accuracy, sensitivity, specificity, precision, and area under the curve (AUC). Using different metrics is necessary since the model might have good results in some metrics and poor performance in others metrics.

Most of the evaluation measures are calculated using the confusion matrix (CM) which is a table of 4 different combination having true and predicted condition (see Figure 1.2). CM 2 by 2 matrix, which report numbers of false positives, false negatives, true positives, and true negatives.

Table 1.2: Structure of the confusion matrix for binary classification

	True Condition	
Predicted Condition	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

1. **Accuracy** defined as the percentage of correctly classified instances and is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100. \tag{1.8}$$

2. **Kappa (Cohen's k-coefficient)** is a measure of degree of agreement between the predictions and the true classes:

$$Kappa = \frac{\text{Accuracy} - \text{Random Accuracy}}{1 - \text{Random Accuracy}}. \tag{1.9}$$

$$\text{Random Accuracy} = \frac{(TN+FP)\times(TN+FN)+(FN+TP)\times(FP+TP)}{(TP+TN+FP+FN)^2}.$$

(1.10)

3. **Specificity** is the proportion of cases predicted to be negative given they are true negatives:

$$Specificity = \frac{TN}{(TN+FP)}. \tag{1.11}$$

4. **Sensitivity** is the proportion of cases predicted to be positive given they are truly positive:

$$Sensitivity = \frac{TP}{(TP+FN)}. \tag{1.12}$$

5. **Balanced Error Rate (BER)** is the average of the errors (wrongly classified) in each class:

$$BER = \frac{1}{2}\left(\frac{FP}{(TN+FP)} + \frac{FN}{(FN+TP)}\right). \tag{1.13}$$

6. **Receiver Operating Characteristic Curve (ROC)** is an important measure for evaluating the accuracy of a statistical, machine, and deep learning models. ROC is a probability curve that indicates the models capability in distinguishing between classes. It is a plot of true positive rate (TPR) against false positive rate (FPR).

7. **Area Under the Curve (AUC)**: is an essential measure that indicates the model prediction reliability.

$$AUC = \frac{1}{2}\left(\frac{TP}{(TP+FN)} + \frac{TN}{(TN+FP)}\right). \tag{1.14}$$

   The AUC is between 0 and 1.

8. **Precision** is the ratio of accurate positive predictions to the total number of positive predictions:

$$Precision = \frac{TP}{TP+FP}. \tag{1.15}$$

9. **Recall** is the ratio of accurate positive predictions to all predicted results:

$$Recall = \frac{TP}{TP + FN}. \tag{1.16}$$

10. **F1 Score** is the harmonic mean of precision and recall.

$$F1Score = \frac{2 * (precision * recall)}{precision + recall}. \tag{1.17}$$

1.21 Problem Statement

The computational methods are still faced with methodological challenges including how to deal with high-dimensionality characterized by a large number of genes or probes and much smaller number of samples requiring the need for dimension reduction (Hasan & Adnan, 2012). In addition, recent technological advances have also led to next generation sequence data which contains a large number of biomarkers and genes associated or not associated to a given disease. Prognosis and predictive models are indispensable tools in the early diagnosis and treatment of a patient with cancer. Gene expression data have the potential to supplement thousands of genes for cancer samples. This allows massively parallel gene expression analysis of human tumors, which made a great excitement in the scientist society (Weigelt et al., 2010; Colombo et al., 2011). Thus there is an urgent need for developing an integrated approach to gene selection in cancer survival studies that jointly utilize both sources of information namely the microarray and RNASeq technologies. Under the classification problem several parametric, non-parametric and semi-parametric statistical methods have been proposed within the last decade, but none has been unanimously accepted as the gold standard. Consequently, in this case, the statistical methods fail to deal when the number of variables p is larger than samples size n (high-dimensional data) to come up with perfect results. In addition, statistical methods rely on many rigid assumptions about the relationships between the inputs and the outcome.

Therefore, due to these two disadvantages, the statistical methods are not very suitable for high-dimensional data. Thus, this makes the machine learning and deep learning methods' predictive power strong and reduces human efforts drastically. In addition, there is a growing interest to investigate ways or methods for homogeneous integration of sequencing and gene expression data which may contribute to the more reliable diagnosis.

1.22 Study Justification

The world is experiencing a burden of non-communicable diseases, including cancer. This study aimed to use gene expression data for cancer prediction and to identify the genes that may help in early detection of cancer. A good understanding of cancer nature and predictors that discriminate between the cancerous and normal patient to improve screening, early detection, management, and inform clinical decisions towards improving patient care.

The study focused on cancer diagnosis using gene expression data, which may help in early detection, reduces incidence, increases patient survival, and decreases the impact of cancer-related consequences. Also, analysis of such data leads and supports the physician in learning about the genes, proteins, and other related and unique factors to the patient and tumors. Consequently, this helps to specify the best and most effective treatment and dosage. Cancer patients may, therefore, receive specific therapy based on their gene profile, which leads to improved lives and lowers their hazard of death.

Furthermore, the joint analysis of microarray and RNASeq data in this study may increase research to attain more robust statistical significant results for identifying more reliable biomarkers that may improve patients' survival and quality of life. This study applied relevant machine and deep learning compared to statistical methods that might fail in high-dimensional data. The methods applied in this work provide a profound foundation for applying artificial intelligence in cancer classification.

1.23 Study Objectives

The broad aims of the project are to:

1. To compare ML and DL with the statistical methods.
2. To integrate microarray and RNASeq gene expression data.
3. Identify which methods (including a combination of methods) are a state of the art (baseline method) in discriminating cancer types and sub-types and stages of disease.
4. To determine the genes that are predictors of survival of cancer patients and to examine the influence of sex, age at diagnosis, stages and the molecular sub-types among other clinical information.

1.24 Methodological Approach

Many statistical and machine learning methods have been applied in order to get most relevant genes/ biomarkers for diagnosis and prognosis from omics data. The methodology we will follow is first is to collect or download the gene expression and sequence dataset from the gene expression omnibus (GEO) or the cancer genome atlas (TCGA) respectively. Thereafter, integrate both types of data. Then apply preprocessing such as normalization, transformation, informative genes selection, etc. for the each and the integrated data. Furthermore, we use data reduction methods such as LASSO, and the convolutional neural network (CNN) which are among the recent in deep learning to extract the most suitable features that make the data ready for analysis. In addition, we suggest combining of the feature selection methods (hybrid approach) as a powerful tool that may give accurate and better result than a single approach. Thereafter, assessing our feature selection approach based on classification or predication for the cancer disease clinical stages using a classification model such as multivariate support vector

machine, artificial neural network, naive Bayes, bagging trees, random forest, k-nearest neighbour, and 1D-CNN.

1.25 Contribution to Knowledge

The main contribution to knowledge will be the development of integrated or hybrid models for gene expression data with mutational assessment capacity to predict clinical outcomes. Model combination has proved a powerful tool in other fields of applications where high-dimensional data is prevalent such is in finance. Through this approach the researchers will be able to expand model capabilities beyond gene expression analysis into DNA sequence analysis and single cell RNASeq which is still an active area of research.

The rest of the thesis is organized as follows: Chapter 2 - 4 presents background, methods, results, and discussion based on each paper/manuscript; Lastly, Chapter 5 focus on the general discussion, conclusion, and recommendations.

Chapter 2

Paper I: Colorectal Cancer Classification and Survival Analysis Based on an Integrated RNA and DNA Molecular Signature

This chapter addresses the following objectives:

1. Use of the feature selection techniques to select genes from data sets on the microarray platform.

2. Compare the classification, predictive and prognostic properties of the genes from the microarray platform with genes obtained from matched samples from the RNASeq platform.

3. Compare the classification, predictive and prognostic properties of the genes from the combined hybrid list of genes from both platforms.

4. Use survival analysis methods (traditional and modern) to check the effect of

the genes on the cancer patients.

2.1 Abstract

Background: Colorectal cancer (CRC) is the third most common cancer among women and men in the USA, and recent studies have shown an increasing incidence in less developed regions, including Sub-Saharan Africa (SSA). We developed a hybrid (DNA mutation and RNA expression) signature and assessed its predictive properties for the mutation status and survival of CRC patients.

Methods: Publicly-available microarray and RNASeq data from 54 matched formalin-fixed paraffin-embedded (FFPE) samples from the Affymetrix GeneChip and RNASeq platforms, were used to obtain differentially expressed genes between mutant and wild-type samples. We applied the support-vector machines, artificial neural networks, random forests, k-nearest neighbor, naïve Bayes, negative binomial linear discriminant analysis, and the Poisson linear discriminant analysis algorithms for classification. Cox proportional hazards model was used for survival analysis.

Results: Compared to the genelist from each of the individual platforms, the hybrid genelist had the highest accuracy, sensitivity, specificity, and AUC for mutation status, across all the classifiers and is prognostic for survival in patients with CRC. NBLDA method was the best performer on the RNASeq data, while the SVM method was the most suitable classifier for CRC across the two data types. Nine genes were found to be predictive of survival.

Conclusion: This signature could be useful in clinical practice, especially for colorectal cancer diagnosis and therapy. Future studies should determine the effectiveness of integration in cancer survival analysis and the application on unbalanced data, where the classes are of different sizes, as well as on data with multiple classes.

2.2 Introduction

Colorectal cancer (CRC) is one of the major emerging causes of mortality and morbidity around the world (Gandomani et al., 2017). CRC is also the third leading cause of death among men and women (Siegel et al., 2019; WCRF, 2019; Omolo et al., 2016; Mármol et al., 2017; Granados-Romero et al., 2017). According to the World Health Organization (WHO), there were about 1.80 million new cases and 862,000 deaths in the year 2018 (WHO, 2019). Furthermore, in 2019, CRC was reported to be the third most prevalent cancer among men and women and an estimated 101,420 and 44,180 new cases of colon and rectal cancer, respectively, and 51,020 deaths in the USA alone (Siegel et al., 2019; DeSantis et al., 2019; Society, 2015).

Although the incidence rates of CRC are lower in developing countries than in developed countries, recent studies have shown an increase in the incidence rates in SubSaharan Africa (May & Anandasabapathy, 2019). Many cancer types that are relatively curable in developed countries are detected only at advanced stages in developing countries, due to late or inaccurate diagnoses (Organization, 2002). Cancer tumor classification based on morphological characteristics alone has been shown to have serious limitations in some studies (Golub et al., 1999). Physicians aim to diagnose CRC as early as possible to design optimal treatment strategies that are patient-specific. Therefore, using genetic mutation and features of the tumor would most probably lead to better understanding and early detection of the disease and lead to finding suitable and targeted strategies (Wang et al., 2012).

Previously, most of the cancer classification research was based on clinical features of the tumors, which lacked the accurate diagnostic ability, hence the need to develop new methods that will better address this critical problem (Golub et al., 1999; Tan & Gilbert, 2003). Recently, DNA microarray technology has greatly improved the classification of diseases into sub-types, particularly cancer. This

technology allows the processing of thousands of genes simultaneously, hence providing critical information about a disease (Vanitha et al., 2015; Lusa et al., 2010). Microarray gene expression data have been used widely for cancer detection, prediction, and diagnosis (Rajeswari & Reena, 2011). In the last decade, next-generation sequencing (NGS) technology has emerged as an advancement in cancer and other disease research, based on RNA sequencing methodology. NGS platforms that are most common include Illumina, SOLiD, Ion Torrent semiconductor sequencing, and single-molecule real-time sequencing (Datta & Nettleton, 2014).

NGS technology has been the most attractive, and its application dramatically improved over the last few years. This technology is high-throughput and has become popular in the detection and analysis of differentially expressed genes (Datta & Nettleton, 2014; Rai et al., 2018). More recently, RNASeq data has been shown to be better than microarray data in terms of quality and accuracy in estimating transcript abundance. However, the two methodologies are different in design and implementation (Rai et al., 2018; Castillo et al., 2019; Zhang et al., 2015). Although RNASeq experiments are expensive, in contrast, they have many advantages over microarrays. RNASeq allows detecting the variation of a single nucleotide, does not require genomic sequence knowledge, provides quantitative expression levels, provides isoform-level expression measurements, and offers a broader dynamic range than microarrays (Castillo et al., 2019). Moreover, RNASeq allows the detection of novel transcripts, low background signal, and increased specificity and sensitivity (Dong et al., 2016). However, our view is that integrated use of data from both technologies may be the best approach, given the available information from both technologies.

Microarray and RNASeq technologies produce gene expression data in different forms. The structure of gene expression produced using microarrays is continuous

data, while RNASeq provides a discrete Type of data (Zararsız et al., 2017). What is common between the two technologies is that both generate big datasets consisting of a few sample sizes, where each sample has a large number of genes. Many areas of research, such as clinical, medical, biological, and agriculture, apply the RNASeq technology (Kulski, 2016; Wang et al., 2019b).

Many statistical and machine learning methods have been used to analyze and extract information from massive amounts of gene expression data. These methods include the Poisson linear discriminant analysis (PLDA), negative binomial linear discriminant analysis (NBLDA), support vector machines (SVM), artificial neural networks (ANN), linear discriminant analysis (LDA), and random forests (RF).

These methods have been used and examined in many studies based on RNASeq and microarray data. For example, Aziz et al. (Aziz et al., 2018) assessed the ANN performance based on microarray data using six hybrid feature selection methods. Five gene expression datasets were used for evaluating these methods and for understanding how these methods can improve the performance of ANN. Statistical hypothesis tests were used to check the differences between these methods. They showed that the combination of independent component analysis (ICA) and genetic bee colony algorithm had superior performance. Salem et al. (Salem et al., 2017) proposed a new methodology for gene expression data analysis. They combined information gain (IG) and standard genetic algorithm (SGA) for feature selection and reduction, respectively. Their approach was tested on seven cancer datasets and then compared with the most recent approaches. Their results show that the proposed approach outperformed the most recent approaches. Jain et al. (Jain et al., 2018) presented a two-phase hybrid method for cancer classification using eleven microarray datasets for different cancer types. They combined correlation-based feature selection (CFS) and improved-binary particle swarm optimization (IBPSO). Naive Bayes with 10-fold cross-validation was used for

assessment. Results indicated that their approach had better performance in terms of accuracy and the number of selected genes.

Anders and Huber (Anders & Huber, 2010) conducted differential expression analysis based on the negative binomial distribution, with variance and mean linked by local regression, for count data. Their proposed method controls the Type I error and gives good detection power. Zararsiz et al. (Zararsız et al., 2017) presented a comprehensive simulation study on RNASeq classification using PLDA, NBLDA, single SVM, bagging SVM (bagSVM), classification and regression trees (CART), and RF. Their simulation results were applied and compared to two miRNA and two mRNA real experimental datasets. They found that the power-transformed PLDA, RF, and SVM were the best in classification performance.

Due to the small number of samples for gene expression data, combining independent datasets is novel in order to increase sample size and statistical power. Taminau et al. (Taminau et al., 2014) worked on the integration of gene expression analysis using two approaches based on merging and meta-analysis. They used six gene expression datasets. Results showed that both meta-analysis and merging did well, but merging was able to detect more differentially expressed genes than meta-analysis.

Recently, combining two different gene expression data sources has been shown to improve classification accuracy as opposed to using only one source. Castillo and co-workers (Castillo et al., 2019) introduced the integration of multiple microarrays and RNASeq platforms. They first carried out a differential expression analysis, then applied the minimum-redundancy maximum-relevance (mRMR) feature selection approach for further reduction of the gene-list. The top 10 genes were selected and evaluated using four classification methods: k-nearest neighbor

(KNN), naive Bayes (NB), RF, and SVM. Their results showed the highest accuracy and f1-score for the KNN. In this study, we combined RNASeq and DNA expression data from colorectal cancer patients. We obtained a hybrid gene-list from the RNASeq and microarray datasets and assessed its classification performance based on the PLDA, NBLDA, SVM, RF, ANN, KNN, and NB algorithms.

The paper is structured as follows. Section 2.3 discusses the methods and the datasets used in the study. Section 2.4 shows the classification results of the microarray, RNASeq, hybrid gene lists, and survival analysis. Discussion and conclusions are presented in Sections 2.5 and 2.6, respectively.

2.3 Materials and Methods

2.3.1 Datasets

We used publicly available microarray and RNASeq data that is also reported in Omolo et al. (Omolo et al., 2016). The data consists of 54 matched formalin-fixed paraffin-embedded (FFPE) samples from colorectal cancer patients and is available in the gene expression omnibus (GEO) repository under the accession numbers GSE86562 and GSE86559 for RNASeq and microarray data, respectively. The microarray gene expression data consists of $60,607$ genes on 54 colorectal patients. We used the KRAS mutation status as a class variable. As a first step, the Affymetrix microarray data were log2-transformed and quantile-normalized, and genes with more than 50% missing values were filtered out. After that, we performed class comparison using the two-sample t-test at the 0.005 significant level threshold, which yielded 165 differentially expressed genes.

The RNASeq dataset contained $57,905$ genes from the same colorectal cancer patients used to generate the microarray data. This data is in the form of counts,

i.e., discrete. For this data, first, filtration was done to remove the genes with more than 50% of zeros across the samples, using the counts per million (CPM) method (Lai, 2010). We retained genes whose CPM values are greater than 0.5. Thus, the dimension reduced to $17,473$ genes. We performed differential expression analysis using the DESeq2 package in *R*. This step reduced the genes to 282 genes using the 0.005 significance threshold level. The differential expression analysis tool in DESeq2 uses a generalized linear model (GLM) of the following form:

$$g_{ij} \sim NB(\mu_{ij}, \alpha_i), \;\; \mu_{ij} = s_j q_{ij}, \;\; log_2(q_{ij}) = x_{j.}\beta_i, \tag{2.1}$$

where g_{ij} is the counts for gene i in sample j. These counts are modeled using a negative binomial distribution with fitted mean μ_{ij} and a gene-specific dispersion parameter α_i. The fitted mean is decomposed into a sample-specific size factor s_j and a parameter q_{ij} proportional to the expected true concentration of fragments for sample j. The $x_{j.}$ is the feature or vector of features associated with sample j. The coefficients β_j represent the log2-fold changes for gene i for each column of the model or design matrix $\boldsymbol{X}$. Note that the model can be generalized to use a sample- and gene-dependent normalization factors s_{ij}.

The dispersion parameter α_i defines the relationship between the variance of the observed count and its mean value. That is, how far we expect the observed count to be from the mean value, which depends both on the size factor s_j and the covariate-dependent part q_{ij} as defined above. Thus, the variance function is given by:

$$Var(g_{ij}) = E[(g_{ij} - \mu_{ij})^2] = \mu_{ij} + \alpha_i \mu_{ij}^2. \tag{2.2}$$

The steps performed by the DESeq function in DESeq2 package are the estimation of s_j, and α_i, and fitting negative binomial GLM for β_i and Wald statistics by *nbinomWaldTest*.

We computed counts per million as:

$$CPM_i = \frac{g_i}{N} * 10^6, \tag{2.3}$$

where g_i denotes the counts observed from a gene of interest i, and N is the number of sequenced fragments.

RNASeq and microarray data integration may help improve cancer classification accuracy. Several studies have addressed the classification problem using RNASeq, microarray, or a combination of both, based on heterogeneous samples (Castillo et al., 2019, 2017; Gomez-Cabrero et al., 2014). Our study aimed to integrate homogeneous samples from the RNASeq and microarray platforms. In this regard, we obtained the differentially expressed genes from the two platforms based on the same set of samples. After that, we used the database for annotation, visualization, and integrated discovery (DAVID) (Huang et al., 2007) and catalogue of somatic mutations in cancer (COSMIC) tools, to annotate the RNASeq transcripts list. The microarray genes symbol names were obtained from the dataset in (Omolo et al., 2016). We then obtained the intersection, complement of the intersection, and union between the two annotated lists.

Integration was done using the intersection, complement of the intersection, and the union of the two lists of genes. Due to the different nature of the two datasets, RNASeq was log2 transformed and quantile-normalized to make both types of data consistent with each other. Subsequently, the integration was done based on binding the two gene-lists from the RNASeq and microarray datasets. To transform the RNASeq data, we let:

$$\text{Transformed Data} = log_2(G + 1), \tag{2.4}$$

where G is the RNASeq counts data matrix, and $G + 1$ is the RNASeq counts data

matrix with all zero counts changed to one.

Quantile normalization ensures that probe intensities of each array in a set of arrays have the same distribution. A quantile-quantile plot would help to confirm if two probe vectors have the same distribution (quantiles lie on the diagonal line) or not. This approach can be extended to n-dimensional data. Let $\mathbf{q}_k = (q_{kn}, \ldots, q_{kn})', k = 1, \ldots, P$, be the vector of the k^{th} quantiles for all n arrays, and $\mathbf{d} = (\frac{1}{\sqrt{n}}, \ldots, \frac{1}{\sqrt{n}})'$ be the unit diagonal. To transform from the quantiles so that they all lie along the diagonal, we projected $\mathbf{q}$ on to $\mathbf{d}$ as below (Bolstad et al., 2003):

$$Proj_{dqk} = (\frac{1}{n}\sum_{j=1}^{n} q_{1j}, \ldots, \frac{1}{n}\sum_{j=1}^{n} q_{Pj}). \tag{2.5}$$

2.3.2 Data Integration

Here, we used homogeneous data from matched-pair samples from microarray and RNASeq technologies. Using a set-theoretic approach of taking the intersection, the complement of the intersection, or union, we obtained four lists of genes from the two platforms at the 0.005 significance level. The intersection between the two lists was 23 genes, with 401 genes being the complement of the intersection. The steps followed in this study are as shown at Figure 2.1.

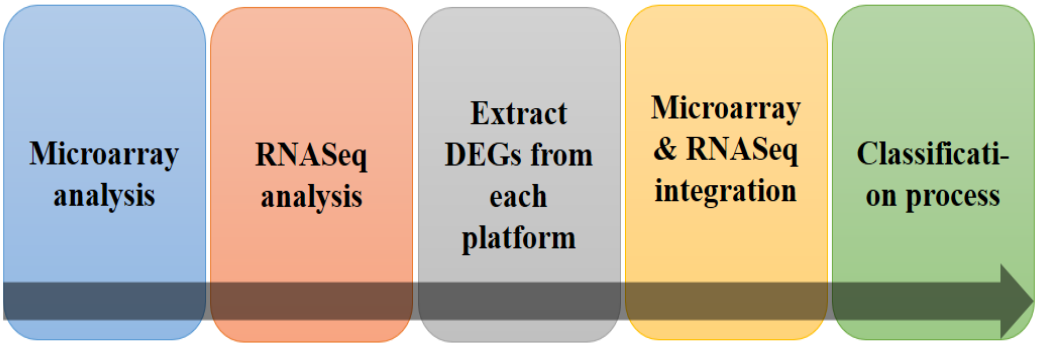


Figure 2.1: Flow-chart of the analysis.

Moreover, the specific steps/methods and packages that have been used in this

study explained in detail in Figure 2.2 below.

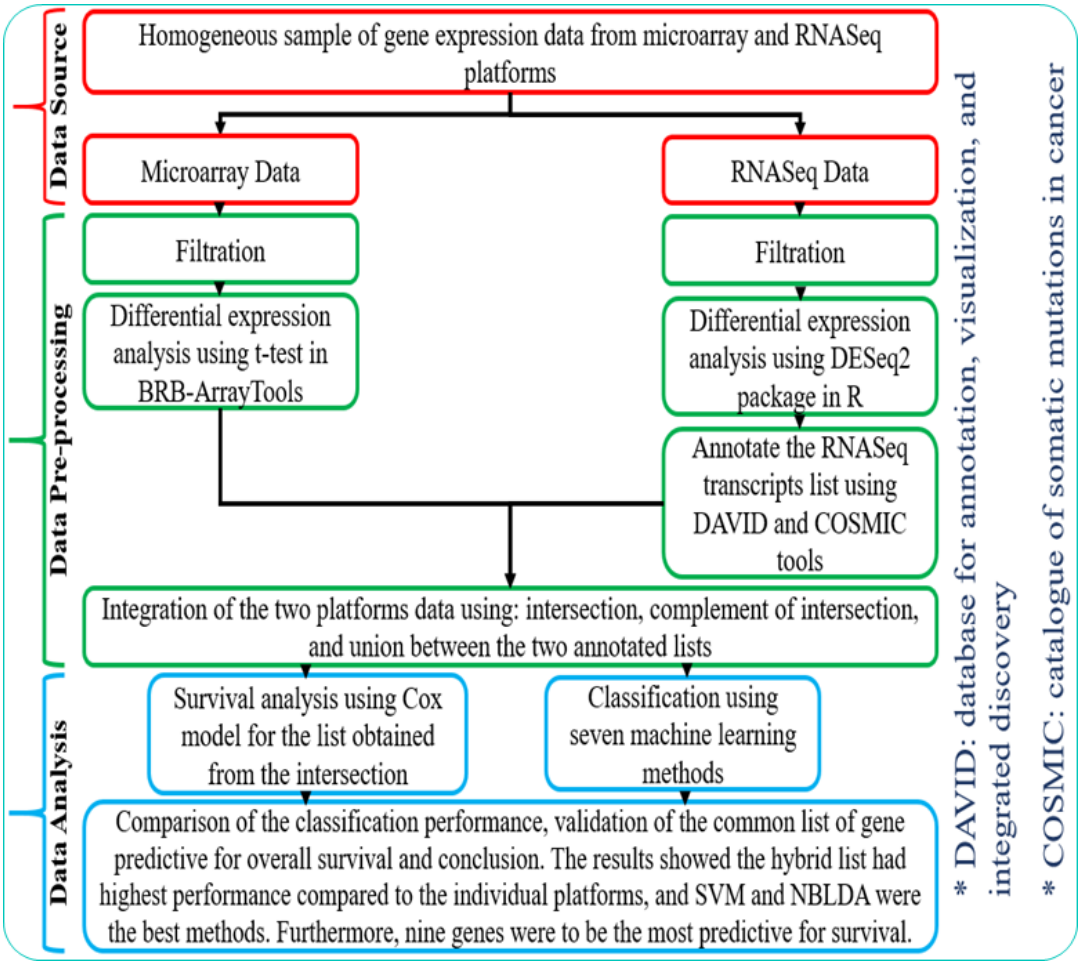


Figure 2.2: Illustrate the steps/methods and packages that used in the study.

2.3.3 Classification Methods

Several methods have been developed for classification and their performance evaluated in both microarray and RNASeq platforms. Below, we briefly describe seven classification methods and how to evaluate their performances based on the integration of the two platforms.

2.3.3.1 Poisson Linear Discriminant Analysis

The PLDA classifier was proposed by Witten (Witten, 2011). Witten used the Poisson log-linear model and developed an analog of diagonal linear discriminant

analysis for sequence data.

Let $\mathbf{G}$ denote a $n \times p$ matrix of read counts data, where n denotes the number of observations (samples), and p the number of genes. Let G_{ij} be the counts or reads for gene j in sample i; it is reasonable to assume that:

$$G_{ij} \sim Poisson(\mu_{ij}), \tag{2.6}$$

where $\mu_{ij} = s_i g_j$. To avoid identifiability issues, one can require $\sum_{i=1}^{n} s_i = 1$, where s_i is the number of counts per sample i, and g_j is the number of counts per gene j.

Suppose that we have K different classes of samples. Then we can write

$$G_{ij}|y_i = K \sim Poisson(\mu_{ij} d_{kj}), \tag{2.7}$$

where y_i denotes the class of the i^{th} sample (y_i=$1, 2, 3, \ldots, K$) and d_{kj} denotes a measure of the level of the i^{th} gene to be differentially expressed in class k.

Let $g_i = (g_{i1}, g_{i2}, \ldots, g_{ip})'$ indicate the entries of row i in the $\mathbf{G}$ matrix, which are the gene expression levels of sample i. Let, $G_{.j} = \sum_{i=1}^{n} G_{ij}$, $G_{i.} = \sum_{j=1}^{p} G_{ij}$ and $G_{..} = \sum_{i,j} G_{ij}$ denote the column, row, and the overall totals, respectively. The maximum likelihood estimate (MLE) for μ_{ij} assuming independence is $\hat{\mu_{ij}} = \frac{G_{i.} G_{.j}}{G_{..}}$, and $\sum_{i=1}^{n} \hat{s}_i = 1$ yields the estimates $\hat{s}_i = \frac{G_{i.}}{G_{..}}$ and $\hat{g}_j = G_{.j.}$ $\hat{s}_i$ is the estimate of the size factor for sample i. Maximum likelihood estimation provides the estimate of d_{kj} as $\hat{d_{kj}} = \frac{G_{c_k j}}{\sum_{i \in c_k} \hat{\mu_{ij}}}$, where c_k denotes the class of an observation.

If $\hat{d_{kj}} > 1$, then the j^{th} gene is over-expressed relative to the baseline in the k^{th} class, and if $\hat{d_{kj}} < 1$, then the j^{th} gene is under-expressed relative to the baseline in the k^{th} class. If $G_{c_k j} = 0$ (an event that is not unlikely if the true mean for j^{th} gene is small), then the maximum likelihood estimates for d_{kj} equals zero.

Assume that we want to classify a new observation $g^* = (G_1^*, \ldots, G_p^*$, and let y^* indicate the unknown class label. By Bayes rule,

$$P(y^* = k|g^*) \propto f_k(g^*)\pi_k, \tag{2.8}$$

where f_k is the density of a sample in class k and π_k is the prior probability that an observation belongs to class k. Then, if f_k is a normal density with a class-specific mean and common covariance, PLDA classifies a new sample to class k, which maximizes equation (2.8). Consequently, the discriminant score of PLDA is

$$log\, Pr(y^* = k|g^*) \approx \sum_{j=1}^{P} G_j^*\, log\, d_{kj} - \sum_{j=1}^{P} s^*\, \lambda_j\, d_{kj} + log\, \pi_k + C. \tag{2.9}$$

PLDA is implemented using the *R* package *MLSeq*.

2.3.3.2 Negative Binomial Linear Discriminant Analysis

Recently, Dong et al. (Dong et al., 2016) proposed NBLDA for RNASeq data analysis. NBLDA and Poisson linear discriminant analysis (PLDA) were considered the most suitable classifiers for RNASeq data due to the discrete nature of data (Dong et al., 2016; Witten, 2011).

Let G_{ij} denote the number of reads in sample i, and gene j, $i = 1, 2, 3, \ldots, n$ and, $j = 1, 2, 3, \ldots, p$. Then G_{ij} is assumed to follow the negative binomial distribution

$$G_{ij} \sim NB(\mu_{ij}, \phi_j),\ \mu_{ij} = s_i\lambda_j, \tag{2.10}$$

where s_i is the size factor, used to scale gene counts for the ith sample due to different sequencing depth, λ_j is the total number of reads per gene, and $\phi_j \geq 0$ is the dispersion parameter. The mean and variance of the negative binomial distribution are given by:

$$E(G_{ij}) = \mu_{ij} \text{ and } V(G_{ij}) = \mu_{ij} + \mu_{ij}^2\phi_j. \tag{2.11}$$

Suppose that we have M classes. Let C_m be an indicator variable such that $C_m \in 1, 2, 3, \ldots, M$. Then, the model for RNASeq data is

$$(G_{ij}|y_i = m) \sim NB(\mu_{ij}d_{m,j}, \phi_j), \tag{2.12}$$

where d_{mj} denotes the differences among the M classes, and $y_i = m, m \in 1, 2, 3, \ldots, M$ denotes the class of samples i. The assumption is that all the genes are independent.

Let $\mathbf{g}^* = (G_1^*, \ldots, G_p^*)$ be a new sample whose class is to be predicted, s^* is the size factor, and y_i^* the class label value. By Bayes' rule, we have

$$Pr(y^* = m|g^*) \propto f_m(g^*)\pi_m, \tag{2.13}$$

where f_m is the pdf of the sample in class m, and π_m is the prior probability that a sample comes from class m. The pdf of $G_{ij} = g_{ij}$ in equation (2.12) is

$$Pr(G_{ij} = g_{ij}|y_i = m) = \frac{\Gamma(g_{ij}+\phi_j^{-1})}{g_{ij}^2\Gamma(\phi_j^{-1})}\left(\frac{s_i\lambda_j d_{mj}\phi_j}{1+s_i\lambda_j d_{mj}\phi_j}\right)^{g_{ij}}\left(\frac{1}{1+s_i\lambda_j d_{mj}\phi_j}\right)^{\phi_j^{-1}}. \tag{2.14}$$

Thus, the discriminant score for NBLDA can be constructed from equations (2.13) and (2.14) as

$$\begin{aligned} log\, Pr(y^* = m|g^*) = &\sum_{j=1}^{P} G_j^*[log\, d_{mj} - log(1 + s_i\lambda_j d_{mj}\phi_j)] \\ &- \sum_{j=1}^{P} \phi_j^{-1}\, log(1 + s_i\lambda_j d_{mj}\phi_j) + log\, \pi_m + C, \end{aligned} \tag{2.15}$$

where C is a constant independent of m. The class m, which maximizes the score in

equation (2.15) will be assigned to the new sample g^*. NBLDA is implemented using the R package MLSeq.

2.3.3.3 Support Vector Machines

The SVM method was first proposed by Boser, Guyon, and Vapnik (Boser et al., 1992) at the Computational Learning Theory (COLT92) ACM Conference in 1992. The method is based on the idea of a hyperplane that lies furthermost from both classes. This plane is known as the optimal (maximum) margin hyperplane. The hyperplane is completely determined by a sub-set of the samples known as the support vectors (Moguerza & Muñoz, 2006). SVM has the ability to handle problems where the data are not linearly separable by transforming the data using mapping kernel functions such as the radial basis function (RBF) kernel, polynomial function, and the linear function (Stephens & Diesing, 2014). In addition, SVM can handle high-dimensional data, which is an essential advantage in dealing with genetic data from cancer studies. This attribute makes SVM widely appealing and applicable to real-life data analysis problems such as handwritten character recognition, human face recognition, radar target identification, speech identification, and, quite recently, to gene expression data analysis (Brown et al., 1999; Chu & Wang, 2003).

Suppose we have n samples and p genes. Further, assume samples belong to two distinct outcome classes represented by $+1$ or -1 and a feature vector $\mathbf{g}_i$ such that $(\mathbf{g}_i, y_i) \in G \times Y$, $i = 1, 2, \ldots, n$, where $\mathbf{g}_i = (g_{i1}, g_{i2}, \ldots, g_{ip})'$ is the sample profile (vector) and $y_i \in +1, -1$ is the outcome class dichotomy. The goal is to classify the samples into one of the two classes by training the SVM which maps the input data (using a suitable kernel function) onto a high-dimensional space (feature space) $\{(\Phi(\mathbf{g}_i), y_i)\}_{i=1}^n$. This is achieved by constructing an optimal separating hyperplane that lies furthest from both classes.

The general form of a separating hyperplane in the space of the mapped data is

defined by

$$\mathbf{W}^T\Phi(\mathbf{g}) + b = 0. \tag{2.16}$$

Here, $\mathbf{W} = (W_1, W_2, \ldots, W_n)'$ is the weight vector. We can rescale the W and b such that the following equation determines the point in each class that is nearest to the hyperplane defined by the equation:

$$|\mathbf{W}^T\Phi(\mathbf{g}) + b| = 1. \tag{2.17}$$

Therefore, it should follow that for each sample i, $i \in 1, 2, \ldots, n$,

$$\mathbf{W}^T\Phi(\mathbf{g}) + b = \begin{cases} \geq 1, & \text{if } y_i = +1. \\ \leq -1, & \text{if } y_i = -1. \end{cases} \tag{2.18}$$

After the rescaling, the distance from the nearest point in each class to the hyperplane becomes $\frac{1}{\|W\|}$. Thus, the distance between the two classes is $\frac{2}{\|W\|}$, which is called the margin. The solution of the following optimization problem is obtained to maximize the margin:

$$\underset{W,b}{min}\|\mathbf{W}\|^2 \text{ subject to } y_i(\mathbf{W}^T\phi(g_i) + b) \geq 1,\ i = 1, 2, \ldots, n. \tag{2.19}$$

The square of the norm of W is considered to make the problem quadratic. Suppose $\mathbf{W}^*$ and b^* are the solutions to the optimization problem 2.19 above. Then this solution determines the hyperplane in the feature space where $(\mathbf{W}^*)^T\Phi(\mathbf{g}) + b^* = 0$. The points $\Phi(\mathbf{g}_i)$ that satisfy the qualities $y_i((\mathbf{W}^*)^T\Phi(\mathbf{g}_i) + b^*) = 1$ are called *support vectors* (Moguerza & Muñoz, 2006). The SVM method is implemented using the *R* package kernlab (Karatzoglou et al., 2019).

2.3.3.4 Random Forests

Random forests were first introduced in 2001 (Hastie et al., 2001; Breiman, 2001b). They are an extension of classification and regression trees, and also an

improvement over bagged trees by further modification using a random small tweak to de-correlate the trees. Growing random forests leads to an improvement in prediction accuracy compared to single or bagged trees (Qi, 2012).

We build a number of forests of decision trees on bootstrapped training samples from the original data. A tree is obtained by recursively splitting the genes such that at each node of the tree, a candidate gene for splitting is obtained from a random sample of size v. A typical choice for v is such that $v = \sqrt{p}$, where p is the number of candidate genes for splitting.

We then grew the trees to maximum depth. Therefore, the two-step randomization process helps to de-correlate the trees (Chen & Ishwaran, 2012). To determine the prediction for an unknown sample, an average over all the trees is taken for a regression problem and a majority vote for a classification problem (Hastie et al., 2001; Pappu & Pardalos, 2014; Xu et al., 2012). Random Forest Algorithm for Regression or Classification (Hastie et al., 2001) can be implemented as follows

1. For $b = 1$ to B (# random-forest trees):
   (a) Draw a bootstrap sample of size N from the training data.
   (b) Grow a random-forest tree, T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size, n_{min}, is reached.
   (c) Select v genes at random from the p genes.
   (d) Pick the best gene to split on among the v based on an impurity measure.
   (e) Using the selected gene, split the node into two daughter nodes.
2. To predict a new sample x : let $\hat{C}_b(x)$ be the class prediction of the b-th random-forest tree. Then $\hat{C^B_{rf}}(x) =$ majority vote $\{\hat{C}_b(x)\}_{b=1}^{B}$.

RF is implemented using the R package randomForest (RColorBrewer & Liaw, 2018).

2.3.3.5 Artificial Neural Networks

Artificial neural networks (ANN) are multi-layered models that are constructed from three layers, each layer consisting of nodes called neurons (Dwivedi, 2018). The input layer contains nodes whose number is based on the input features. The output layer contains nodes equal to the number of classes, and finally, the hidden layer contains nodes determined by the level of tuning required. The inputs are weighted by multiplying each input by weight as a measure of its contribution. The layers are connected together via connection weights. These weights are determined through stages of model fitting. The hidden nodes receive the sum weighted from the input layer plus some bias. This summation is passed onto the transform function (activation function) to generate the results. These results are called outputs and interpreted as a class probability in our case.

There are many types of architecture of ANN. Neural networks are used widely in different fields, such as prediction in time series models, economic modeling, and medical applications (Stephens & Diesing, 2014). Also, ANN can be applied to the classification problem using microarray gene expression data (Dwivedi, 2018). In this paper, we apply the method to both microarray and RNA Sequencing gene expression data.

Consider the simplest multi-layered network with one hidden layer. Assume we have gene expression data where denotes the number of genes. Then the input layer receives the gene expression levels for a sample, each multiplied by the corresponding weight, $W_{ij}^{(1)} g_i$, as shown in equation (2.20), below:

$$b_i = \sum_{j=0}^{n} W_{ij}^{(1)} g_i \quad i = 1, 2, \ldots, m, \tag{2.20}$$

where $\mathbf{g} = (g_0, g_1, g_2, \ldots, g_n)'$ is a vector of input features and $g_0 = 1$ is a constant input feature that with weight W_{i0}. The quantities, b_i, are called *activations*, and the parameters $W_{ij}^{(1)}$ are the weights. Note that alternatively b_i can be viewed as a

summary of the n genes from sample i. The "(1)" superscript indicates that this is the first layer of the network. Each of the activations is then transformed by a nonlinear activation function , typically a sigmoid, as in equation (2.21) below:

$$z_i = f(b_i) = \frac{1}{1 + exp(-b_i)}. \tag{2.21}$$

The quantities z_i are interpreted as the output of hidden units, so-called because they do not have values specified by the problem (as is the case for input units) or target values used in training (as is the case for output units).

In the second layer, the outputs of the hidden units are linearly combined to give the activations:

$$a_k = \sum_{i=0}^{m} W_{ik}^{(2)} z_i \quad k = 1, 2, \ldots, K. \tag{2.22}$$

Again, $z_0 = 1$ corresponds to the bias. Weights $W_{ik}^{(2)}$ parameterize the transformations in the second layer of the neural network. The output units are transformed using an activation function. Again, a sigmoid function may be used as shown below:

$$y_k = f(a_k) = \frac{1}{1 + exp(-a_k)}. \tag{2.23}$$

These equations may be combined to give the overall equation describing the forward propagation through the network, and describes how an output vector is computed from an input vector, given the weight matrices as:

$$y_k = f\left(\sum_{i=0}^{m} W_{ik}^{(2)} f\left(\sum_{j=0}^{n} W_{ik}^{(1)} g_i\right)\right). \tag{2.24}$$

ANN is implemented using the R package nnet (Ripley et al., 2016).

2.3.3.6 Naïve Bayes

The Naive Bayes classifier uses probability theory to find the most likely of the possible classes in a classification problem. The NB classifier relies on two assumptions, namely, that each attribute is conditionally independent of the other attributes given the class and that all the attributes have an influence on the class (De Campos et al., 2011). The popularity of this classifier is mainly due to its simplicity, yet exhibiting a surprisingly competitive predictive accuracy. The NB classifier has previously been applied in many fields, including microarray gene expression data (Stephens & Diesing, 2014; Dwivedi, 2018).

Consider an $n \times p$ gene expression data matrix, where n is the number of the samples, and p is the number of the genes (features). Let g_{kj}, $j = 1, 2, \ldots, p$, denote the j^{th} gene on the k^{th} sample. Let $\mathbf{C}_i$ be the i^{th} class, $i = 1, 2, 3, \ldots, L$. The Naive Bayes classifier uses the *maximum a posteriori* (MAP) classification rule to classify these samples. The probability of the k^{th} sample gene information vector, $\mathbf{G}_k = (g_{k1}, g_{k2}, \ldots, g_{kp})^{'}$, is calculated, and then the sample is assigned the class with the largest probability from L conditional probabilities.
Let $P(C_1|\mathbf{G}_k), P(C_2|\mathbf{G}_k), \ldots, P(C_L|\mathbf{G}_k)$ denote the set of L conditional probabilities. The NB classification depends on the Bayes rule, which states that a posterior probability:

$$P(C_i \,|\, \mathbf{G}_k) = \frac{P(\mathbf{G}_k \,|\, C_i)P(C_i)}{P(G_k} \propto P(\mathbf{G}_k \,|\, C_i)P(C_i), \quad k = 1, 2, \ldots, n, \tag{2.25}$$

where $P(\mathbf{G}_k)$ is considered a common normalizing factor for all the L probabilities. The NB classification assumes that all input features are conditionally independent, that is,

$$\begin{aligned} P(g_{k1}, g_{k2}, \ldots, g_{kp} \,|\, C_i) &= P(g_{k1} \,|\, g_{k2}, \ldots, g_{kp}, C_i) P(g_{k2}, \ldots, g_{kp} \,|\, C_i) \\ &= P(g_{k1} \,|\, C_i) P(g_{k2}, \ldots, g_{kp} \,|\, C_i) \\ &= P(g_{k1} \,|\, C_i) P(g_{k2} \,|\, C_i) \ldots P(g_{kp} \,|\, C_i). \end{aligned} \tag{2.26}$$

Ultimately, NB classifies a new sample, $\mathbf{G}^*$, according to the model with MAP probability given the sample, as

$$\text{Class}(\mathbf{G}^*)_{MAP} = \text{argmax}(P(C_i \,|\, \mathbf{G}^*)). \tag{2.27}$$

NB is implemented using the *R* package naivebayes.

2.3.3.7 *k*-Nearest Neighbors

The k-nearest neighbor classifiers (KNN) are known to be the most useful instance-based learners. KNN is a non-parametric model (Ripley et al., 2016). If the classification is based on Euclidean distance in a feature space, then k determines the number of neighbors to be used. In the testing set, the new sample is assigned to the class that is most likely among the k neighbors. Then the number of neighbors can be tuned to choose the optimal fitted model parameters (Stephens & Diesing, 2014; Dwivedi, 2018; Yao & Ruzzo, 2006).

The KNN uses the Euclidean distance measure to find the closest samples for the new sample. Suppose we have two samples, each one with n genes. Denote the two samples as $S_1 = (g_{11}, g_{12}, \ldots, g_{1n})'$ and $S_2 = (g_{21}, g_{22}, \ldots, g_{2n})'$. Then the Euclidean distance is calculated as the square root of the sum of the squared differences in their corresponding values. Using the Euclidean distance formula, the distance between two points, $dist(S_1, S_2)$, is given as:

$$dist(S_1, S_2) = \sqrt{\sum_{j=1}^{n} (g_{1j} - g_{2j})^2}, \tag{2.28}$$

where a large $dist(S_1, S_2)$ means the two samples belong to different classes, and values near zero suggest that the samples are homogeneous. KNN is implemented using the *R* package caret.

2.4 Results

The analysis of RNASeq data using the integrated list of genes was performed using R statistical software. Assessment of the methods was done using 10-fold cross-validation. Here, the 54 CRC samples were divided into 10-folds randomly, with each fold consisting of about 5 - 6 samples. After that, we used a nine-folds for model-building and one-fold for the testing and validation. Thus, this process was self-iterated ten times, and the average of the ten iterations used to obtain the model performance measures. Several performance measures exist in the literature that can assess classification based on microarray and RNASeq gene expression data. The metrics include accuracy, sensitivity, specificity, kappa coefficient, AUC, and balanced error rate (BER) (Tharwat, 2020; Mohammed et al., 2018).

Table 2.1 below provides the number of genes obtained through the intersection, complement of intersection, and union of the gene-lists from differential expression analysis (RNASeq: $GSE86562$, Microarray: $GSE86559$). There were 165 and 282 total DEGs in the $GSE86559$ and $GSE86562$ datasets, respectively. We obtained 23 genes through the intersection, 142 from a complement of $GSE86559$, 259 a complement of $GSE86562$, and 424 from a union (see Table 2.1).

Table 2.1: The number of genes obtained through the intersection, the complement of intersection, and union of the gene-lists from differential expression analysis (RNASeq: $GSE86562$, Microarray: $GSE86559$).

Dataset	Total of DEGs	Intersection	Complement of Intersection	Union
GSE86559	165	23	142	424
GSE86562	282		259	

The 23 genes obtained from the intersection of the RNASeq and microarray gene

expression data, their official gene symbols, and names are in Table 2.2.

Table 2.2: The official gene symbols and the corresponding gene names.

Ensemble Gene ID	Official Gene Symbol	Name
ENSG00000108511	HOXB6	homeobox B6(HOXB6)
ENSG00000169247	SH3TC2	SH3 domain and tetratricopeptide repeats 2(SH3TC2)
ENSG00000120068	HOXB8	homeobox B8(HOXB8)
ENSG00000025293	PHF20	PHD finger protein 20(PHF20)
ENSG00000136997	MYC	v-myc avian myelocytomatosis viral oncogene homolog (MYC)
ENSG00000143882	ATP6V1C2	ATPase H+ transporting V1 subunit C2(ATP6V1C2)
ENSG00000003096	KLHL13	kelch like family member 13(KLHL13)
ENSG00000131746	TNS4	tensin 4(TNS4)
ENSG00000196532	HIST1H3C	histone cluster 1 H3 family member c(HIST1H3C)
ENSG00000233101	HOXB-AS3	HOXB cluster antisense RNA 3(HOXB-AS3)
ENSG00000204104	TRAF3IP1	TRAF3 interacting protein 1(TRAF3IP1)
ENSG00000126003	PLAGL2	PLAG1 like zinc finger 2(PLAGL2)
ENSG00000120875	DUSP4	dual specificity phosphatase 4(DUSP4)
ENSG00000164070	HSPA4L	heat shock protein family A (Hsp70) member 4 like (HSPA4L)
ENSG00000111057	KRT18	keratin 18(KRT18)
ENSG00000260807	LMF1	lipase maturation factor 1(LMF1)
ENSG00000174136	RGMB	repulsive guidance molecule family member b(RGMB)
ENSG00000197818	SLC9A8	solute carrier family 9 member A8(SLC9A8)
ENSG00000187372	PCDHB13	protocadherin beta 13(PCDHB13)
ENSG00000140526	ABHD2	abhydrolase domain containing 2(ABHD2)
ENSG00000166068	SPRED1	sprouty related EVH1 domain containing 1(SPRED1)
ENSG00000182742	HOXB4	homeobox B4(HOXB4)
ENSG00000101193	GID8	GID complex subunit 8 homolog (GID8)

We performed an exploratory analysis of the RNASeq data. Fig 2.3 shows the most meaningful changes at the 0.005 significance level among the genes between the two conditions, based on the volcano plot (Li, 2012). The volcano plot shows the genes with smaller p-values (higher -log_{10} values) in red.

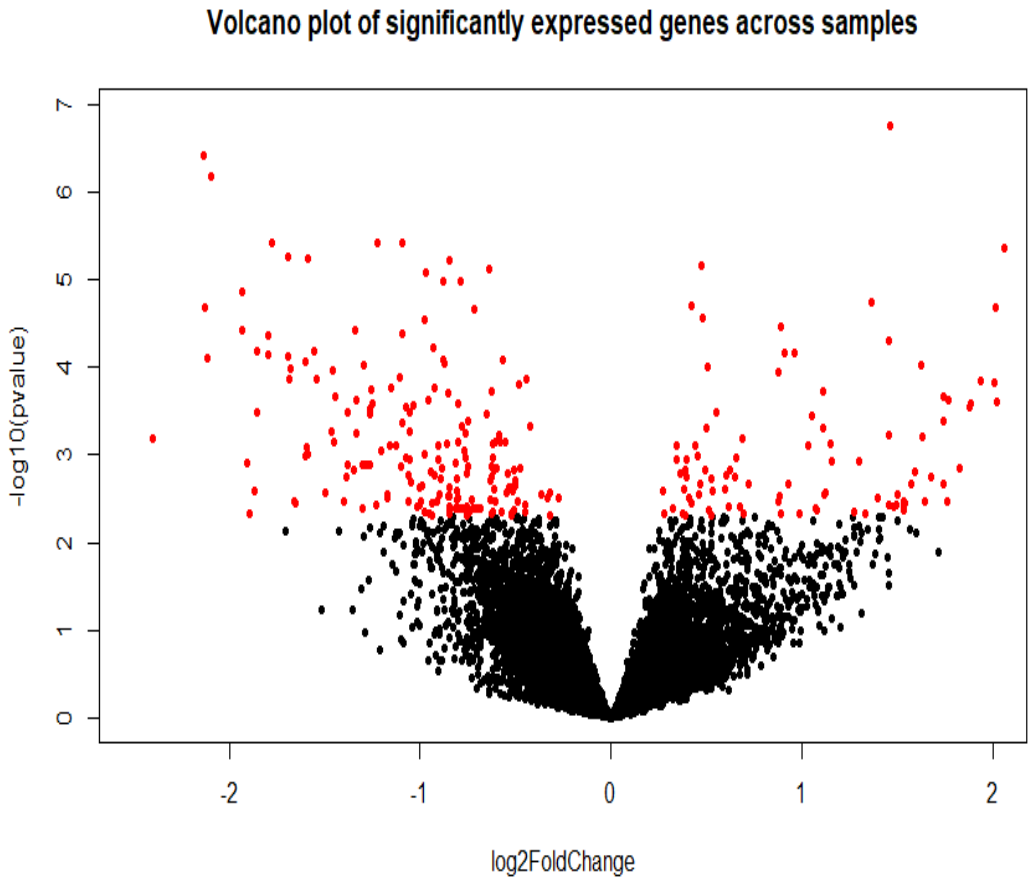


Figure 2.3: Volcano plot of the RNASeq dataset shows the 282 differentially expressed genes in red points ($\alpha = 0.005$).

Fig 2.4 illustrates the estimated dispersion of the RNASeq data using *DESeq2* package, with each gene having a gene-specific dispersion parameter. Good estimates of dispersion parameters lead to accurate detection of differentially expressed genes. Underestimating the dispersion parameters might lead to false positives (i.e., declaring genes to be differentially expressed when they are not truly differentially-expressed). On the other hand, overestimating the dispersion parameters might lead to false negatives (Landau & Liu, 2013).

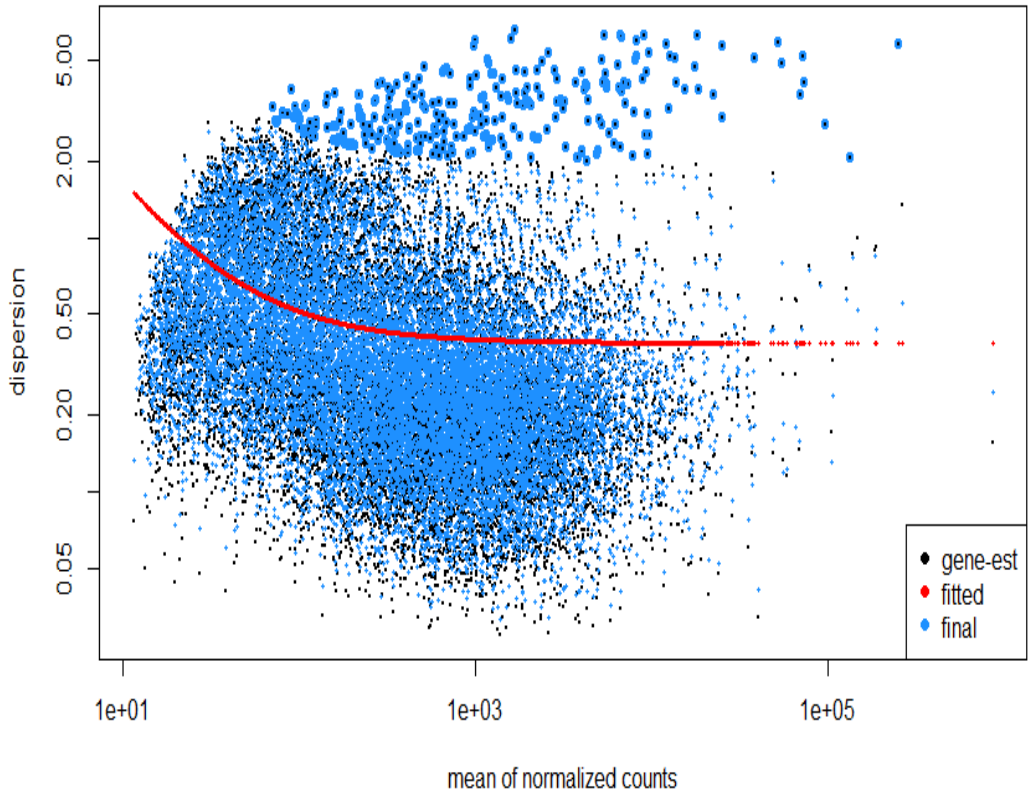


Figure 2.4: Dispersion for the RNASeq data.

Tables 2.3 – 2.6 show the performance of the gene-lists in predicting mutation status, based on seven methods (algorithms), at the 0.005 significance level: the 282 gene-list (Table 2.3); the 23 gene-list (Table 2.4); the 424 gene-list (Table 2.5); and the 401 gene-list (Table 2.6). It is apparent from Table 2.3, compared to Table 2.4 below, that NB, ANN, KNN, and PLDA were improved in the common 23 genes in terms of all performance measures, while RF and NBLDA had the same performance. SVM had a better result on the full list of 282 genes. Therefore, in general, four methods out of seven were improved on the 23 gene-list compared to the 282 genes-list. From Fig 2.5 (a and b), we notice NBLDA works very well in both lists of genes.

Table 2.3: Performance of the classification methods for the 282 gene-list, on the RNASeq dataset ($\alpha = 0.005$).

Metric	Methods						
	SVM	**NB**	**RF**	**ANN**	**KNN**	**NBLDA**	**PLDA**
Accuracy (95% CI)	0.80	0.76	0.83	0.72	0.72	0.89	0.80
	(0.66, 0.89)	(0.62, 0.87)	(0.71, 0.92)	(0.58, 0.84)	(0.58, 0.84)	(0.77, 0.96)	(0.66, 0.89)
Sensitivity (95% CI)	0.89	0.59	0.78	0.78	0.67	0.81	0.81
	(0.71, 0.98)	(0.39, 0.78)	(0.58, 0.91)	(0.58, 0.91)	(0.46, 0.83)	(0.62, 0.94)	(0.62, 0.94)
Specificity (95% CI)	0.70	0.93	0.89	0.67	0.78	0.96	0.78
	(0.50, 0.86)	(0.76, 0.99)	(0.71, 0.98)	0.46, 0.83)	(0.58, 0.91)	(0.81, 1.00)	(0.58, 0.91)
Kappa (95% CI)	0.59	0.52	0.67	0.44	0.44	0.78	0.59
	(0.38, 0.80)	(0.30, 0.73)	(0.47, 0.86)	(0.21, 0.68)	(0.21, 0.68)	(0.61, 0.94)	(0.38, 0.81)
AUC	0.86	0.77	0.87	0.72	0.78	0.94	0.80
BER	0.19	0.21	0.16	0.28	0.28	0.10	0.20

Table 2.4: Performance of the classification methods for the 23 gene-list, on the RNASeq dataset ($\alpha = 0.005$).

Metric	Methods						
	SVM	**NB**	**RF**	**ANN**	**KNN**	**NBLDA**	**PLDA**
Accuracy (95% CI)	0.78	0.80	0.83	0.80	0.76	0.89	0.87
	(0.64, 0.88)	(0.66, 0.89)	(0.71, 0.92)	(0.66, 0.89)	(0.62, 0.87)	(0.77, 0.96)	(0.75, 0.95)
Sensitivity (95% CI)	0.81	0.70	0.78	0.81	0.70	0.85	0.85
	(0.62, 0.94)	(0.50, 0.86)	(0.58, 0.91)	(0.62, 0.94)	(0.50, 0.86)	(0.66, 0.96)	(0.66, 0.96)
Specificity (95% CI)	0.74	0.89	0.89	0.78	0.81	0.93	0.89
	(0.54, 0.89)	(0.71, 0.98)	(0.71, 0.98)	(0.58, 0.91)	(0.62, 0.94)	(0.76, 0.99)	(0.71, 0.98)
Kappa (95% CI)	0.56	0.59	0.67	0.59	0.52	0.78	0.74
	(0.33, 0.78)	(0.38, 0.80)	(0.47, 0.86)	(0.38, 0.81)	(0.29, 0.75)	(0.61, 0.94)	(0.56, 0.92)
AUC	0.80	0.82	0.91	0.84	0.78	0.89	0.91
BER	0.22	0.19	0.16	0.20	0.24	0.11	0.13

Table 2.5: Performance of the classification methods for the 424 gene-list, on the combined RNASeq and microarray datasets ($\alpha = 0.005$).

Metric	Methods				
	SVM	**NB**	**RF**	**ANN**	**KNN**
Accuracy (95% CI)	0.98 (0.90, 1.00)	0.93 (0.82, 0.98)	0.93 (0.82, 0.98)	0.98 (0.90, 1.00)	0.83 (0.71, 0.92)
Sensitivity (95% CI)	1.00 (0.87, 1.00)	0.89 (0.71, 0.98)	0.89 (0.71, 0.98)	1.00 (0.87, 1.00)	0.74 (0.54, 0.89)
Specificity (95% CI)	0.96 (0.81, 1.00)	0.96 (0.81, 1.00)	0.96 (0.81, 1.00)	0.96 (0.81, 1.00)	0.93 (0.76, 0.99)
Kappa (95% CI)	0.96 (0.89, 1.00)	0.85 (0.71, 0.99)	0.85 (0.71, 0.99)	0.96 (0.89, 1.00)	0.67 (0.47, 0.86)
AUC	1.00	0.94	0.96	1.00	0.89
BER	0.02	0.07	0.07	0.02	0.15

Table 2.6: Performance of the classification methods for the 401 gene-list, on the RNASeq dataset ($\alpha = 0.005$).

Metric	Methods				
	SVM	**NB**	**RF**	**ANN**	**KNN**
Accuracy (95% CI)	0.98 (0.90, 1.00)	0.93 (0.82, 0.98)	0.93 (0.82, 0.98)	0.96 (0.87, 1.00)	0.83 (0.71, 0.92)
Sensitivity (95% CI)	1.00 (0.87, 1.00)	0.89 (0.71, 0.98)	0.89 (0.71, 0.98)	0.96 (0.81, 1.00)	0.74 (0.54, 0.89)
Specificity (95% CI)	0.96 (0.81, 1.00)	0.96 (0.81, 1.00)	0.96 (0.81, 1.00)	0.96 (0.81, 1.00)	0.93 (0.76, 0.99)
Kappa (95% CI)	0.96 (0.89, 1.00)	0.85 (0.71, 0.99)	0.85 (0.71, 0.99)	0.93 (0.83, 1.00)	0.67 (0.47, 0.86)
AUC	1.00	0.91	0.96	1.00	0.89
BER	0.02	0.07	0.07	0.04	0.15

Table 2.5 presents the integration results using the union approach, and it is clear that SVM, NB, RF, ANN, and KNN methods were improved compared to the case of 282 differentially expressed genes. Fig 2.5 (a and c) confirm these results. Moreover, SVM and ANN had a higher accuracy than the other methods.

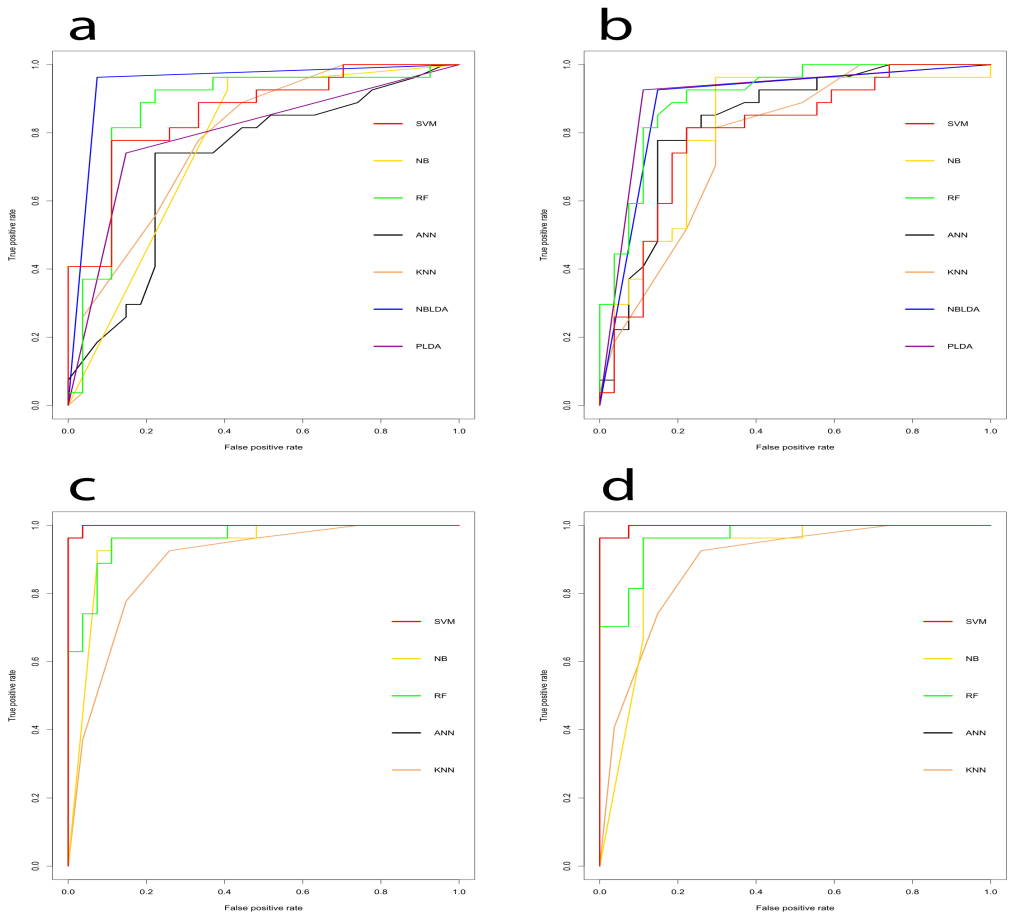


Figure 2.5: ROC curves based on the (a) 282 gene-list for the RNASeq data, (b) 23 gene-list for the RNASeq data, (c) 424 gene-list for the RNASeq and microarray datasets, and (d) 401 gene-list for the RNASeq and microarray datasets, under ($\alpha = 0.005$).

As can be seen from Table 2.6 above, the methods performed better for the gene-list of 401 genes, compared to the 282 gene-list. Furthermore, Fig 2.5 (d) confirm these results.

We compared our gene-list of 23 genes with the 18-gene RAS signature (DUSP4, DUSP6, ELF1, ETV4, ETV5, FXYD5, KANK1, LGALS3, LZTS1, MAP2K3, PHLDA1, PROS1, S100A6, SERPINB1, SLCO4A, SPRY2, TRIB2, and ZFP106) as reported by Dry et al. (Dry et al., 2010) and found only one overlapping gene (DUSP4). It turned out that this was also the most predictive of the seven genes (DUSP4, DUSP6, ETV4, ETV5, PHLDA1, SERPINB1, and TRIB2) that were discussed in Omolo et al. (2016) (Omolo et al., 2016).

We performed an additional analysis to assess whether the 23 gene-list was predictive of overall survival (OS). We used the mutation status as a group variable and vital status (dead or alive) as the censoring variable in this analysis. Overall, there were 20 deaths out of the 54 samples. The results showed that the median OS was 1692 days for the 54 samples. We used the Kaplan-Meier curves to graphically compare survival probabilities (Fig 2.6) between the two mutation groups (RAS-mutant vs. wild-type), and the log-rank test using the RAS mutation status as the group variable. There was no significant difference in OS between the two groups (log-rank = 1.8, p-value = 0.2). We then applied the Cox proportional hazards (CPH) model was to assess the significance of the 23 genes and RAS mutation status. The results show that 9 of 23 genes were significantly associated with OS, including SPRED1, KLHL13, HOXB4, LMF1, HSPA4L at the 0.05 level, and ATP6V1C2, PLAGL2, MYC, SLC9A8 at the 0.1 level (LRT = 56.85, p-value = 0.0002) as can be seen in Table 2.7.

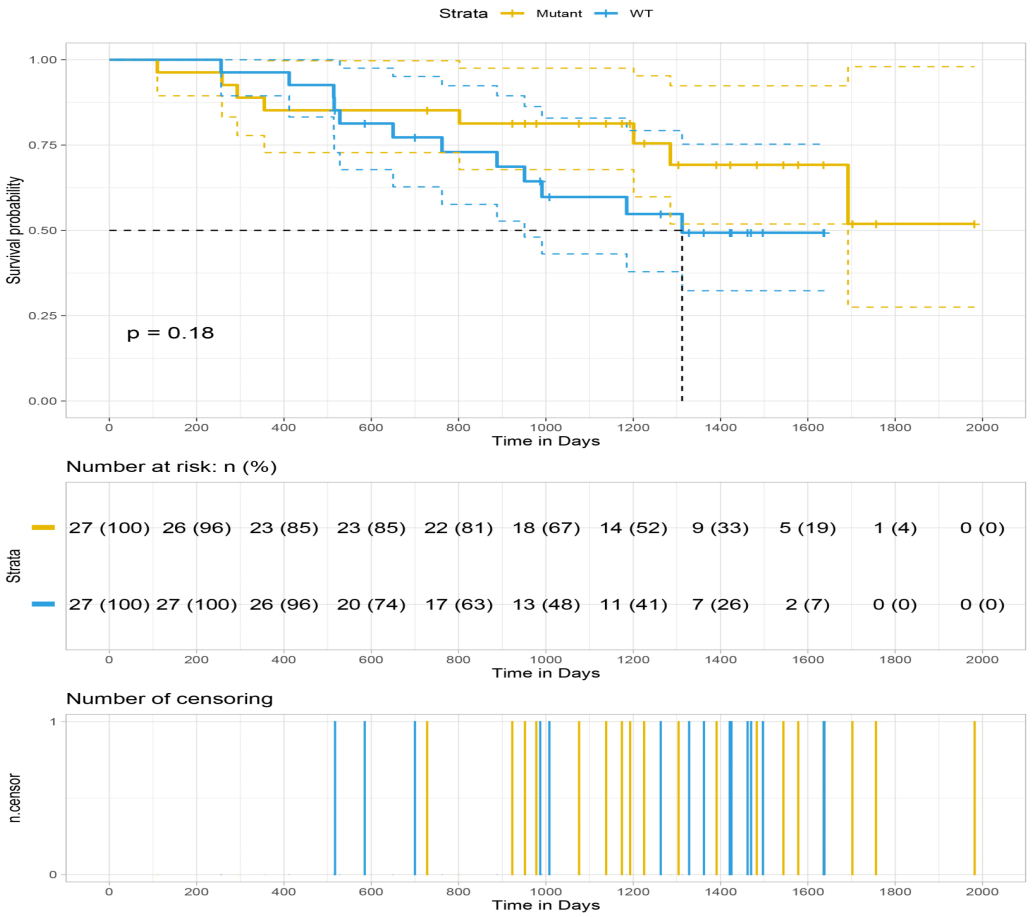


Figure 2.6: Kaplan-Meier curves for overall survival (in days).

We further performed an analysis of the top nine genes using gradient boosted trees and Shapley additive explanations (SHAP) methods to identify the top-K genes $(1 < K < 9)$ (Lundberg & Lee, 2017). The SHAP approach determined the order of importance of our nine genes. SHAP values gave the importance of a gene by comparing what a model predicts with and without the gene. A SHAP value of 0 means that the gene does not affect the prediction, as shown in Fig 2.7. The vertical axis showed the gene names, arranged in the order of importance, from top to bottom while the adjacent value next to the gene name is the mean SHAP value. The horizontal axis showed the SHAP value, which indicated how much the change was in log-odds. From the log-odds, one can obtain the probability of success. The gradient color indicated the original value for that gene.

Genes pushing the prediction higher were colored blue, while those pushing the prediction lower were colored yellow. Each point represented a row from the original dataset.

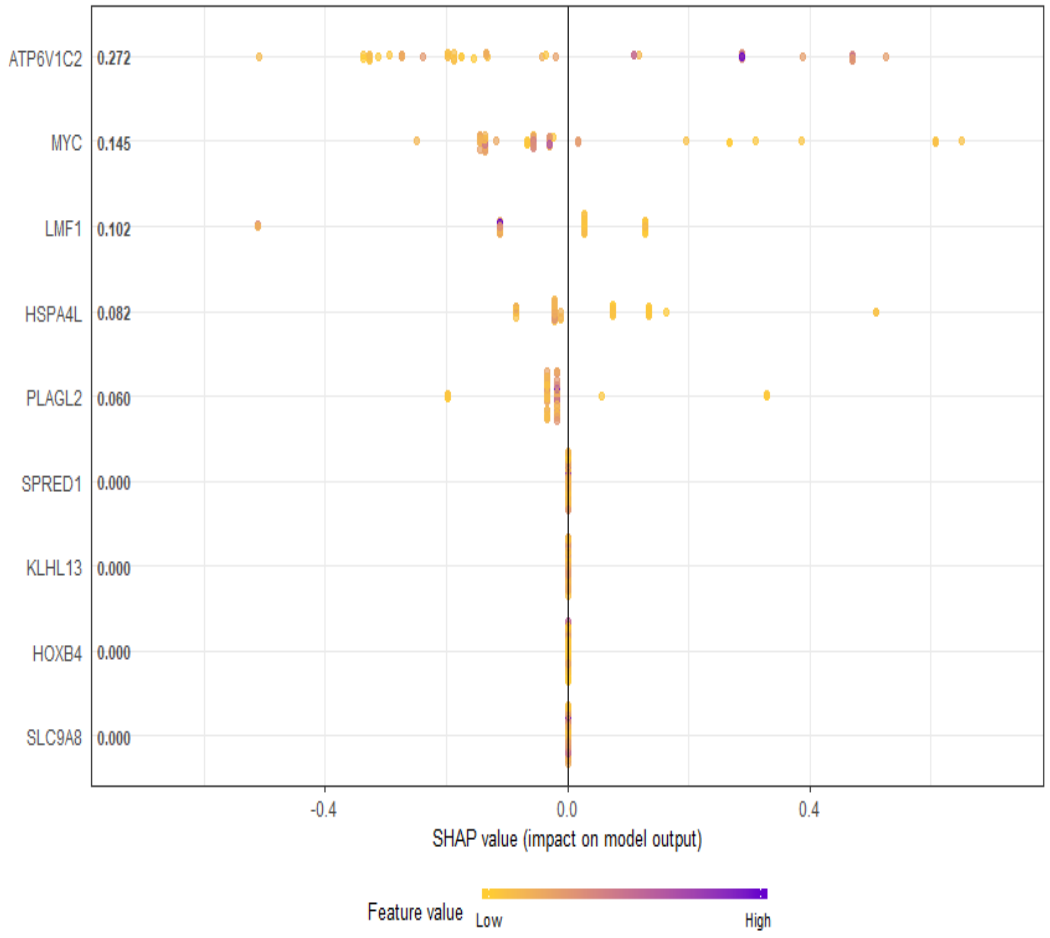


Figure 2.7: Genes in ascending order of importance (Note: dots represent SHAP values of specific features).

Table 2.7: Cox proportional hazards model for overall survival, using the 23 genes and RAS mutation status (class) as covariates.

Covariate	Coef	Hazard ratio (HR)	SE(Coef)	Z-score	P-value
Class	2.23E+00	9.34E+00	1.41E+00	1.589	0.112
ATP6V1C2	4.72E-03	1.01E+00	2.66E-03	1.779	0.0752 .
HOXB-AS3	7.96E-05	1.00E+00	1.30E-03	0.061	0.951
KRT18	-3.27E-05	1.00E+00	8.73E-05	-0.374	0.7084
RGMB	7.48E-04	1.00E+00	1.13E-03	0.662	0.5079
PLAGL2	-3.28E-03	9.97E-01	1.89E-03	-1.737	0.0824 .
DUSP4	2.12E-03	1.00E+00	2.24E-03	0.946	0.3441
SPRED1	1.50E-03	1.00E+00	7.61E-04	1.969	0.0489 *
SH3TC2	6.92E-04	1.00E+00	4.42E-04	1.564	0.1178
HOXB8	-1.04E-02	9.90E-01	7.21E-03	-1.444	0.1488
ABHD2	4.67E-04	1.00E+00	4.34E-04	1.078	0.2812
TNS4	-5.58E-04	9.99E-01	3.74E-04	-1.491	0.1358
HIST1H3C	-4.67E-03	9.95E-01	3.20E-03	-1.458	0.1449
KLHL13	8.28E-03	1.01E+00	3.24E-03	2.559	0.0105 *
MYC	1.21E-03	1.00E+00	7.16E-04	1.689	0.0911 .
HOXB4	9.10E-03	1.01E+00	3.60E-03	2.529	0.0114 *
HOXB6	-2.78E-04	1.00E+00	4.18E-03	-0.067	0.9469
PHF20	1.47E-03	1.00E+00	1.78E-03	0.825	0.4094
LMF1	3.19E-03	1.00E+00	1.43E-03	2.224	0.0262 *
SLC9A8	-4.98E-03	9.95E-01	2.56E-03	-1.946	0.0517 .
GID8	2.16E-03	1.00E+00	2.61E-03	0.828	0.4079
HSPA4L	-6.94E-03	9.93E-01	2.78E-03	-2.497	0.0125 *
PCDHB13	-3.90E-03	9.96E-01	4.09E-03	-0.953	0.3406
TRAF3IP1	-7.63E-03	9.92E-01	4.86E-03	-1.571	0.1163

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

2.5 Discussion

The development of molecular signatures is a significant step towards understanding the molecular mechanisms of tumorigenesis, which could help with accurate prognosis and diagnosis and thus allow physicians to prescribe suitable patient-specific therapies.

Several studies have done cancer classification using either microarray or RNASeq data only, and few have shown integration of both types of data, based on heterogeneous datasets. To the best of our knowledge, no cancer classification study has employed the integration of a homogeneous datasets approach. In this study, we integrated homogeneous microarray and RNASeq datasets and assessed whether such an approach could improve the classification accuracy using seven methods, namely, SVM with radial basis function kernel, NB, RF, ANN, KNN, NBLDA, and PLDA. We implemented the classification of the mutation status of CRC samples, using gene-lists obtained through the intersection, the complement of an intersection, and the union of differentially-expressed genes from microarray and RNASeq datasets.

CRC is the third most common cancer and one of the leading causes of death around the world. The findings suggest that combining two homogeneous datasets from different technologies could lead to an increase in CRC classification accuracy. Castillo et al. (Castillo et al., 2019) reported that combining heterogeneous datasets from different platforms can improve the performance of a classifier, using multiple datasets. They used data from different technologies and platforms to obtain a larger sample size due to the lack of enough RNASeq samples. Our proposed approach is different from Castillo et al. (Castillo et al., 2019), in that we used homogeneous datasets and a balanced binary class problem. We used the 0.005 significance level to obtained the differentially expressed genes, which is restrictive enough to control the false positive rate.

A comparison of the performance of the classification methods for each gene-list revealed that SVM yielded the highest mean accuracy (0.885), followed by RF (0.880), ANN(0.865), NB(0.855), and KNN(0.785) across the four gene-lists. However, NBLDA performed better than PLDA as a classifier when the analysis was restricted to RNASeq (count) data. Castillo et al. (Castillo et al., 2019) also showed that SVM performed second to KNN. Statnikov et al. (Statnikov et al., 2013) performed a comparison of 18 classification methods on five feature selection methods, using eight datasets and showed that RF had the highest accuracy (0.954). Our classification results using the integrated list of genes outperformed Mamatjan et al. (Mamatjan et al., 2017), where they used RNASeq data for tumor classification. Their results showed that mRNA signatures and DNA methylation signatures as single platforms achieved 95% and 88% accuracy of histological diagnosis, respectively. Moreover, the PLAGL2 gene, as one of our predictive genes, also has been one of the most common predictive genes in Rashid et al. (Rashid et al., 2019). Furthermore, our work integrated RNA and DNA signature for classification and survival analysis with very high metrics compared to Popovici et al. (Popovici et al., 2012). In this study (Popovici et al., 2012), the authors developed a classifier using 64 genes for detecting BRAF mutant tumors for colon cancer. Also, they found DUSP4 to be one of the top 50 differently expressed genes.

Survival analysis results showed that 9 of 23 genes were prognostic for overall survival for CRC patients. Upon subjecting the nine genes to the Shapley additive explanations (SHAP) method to rank the genes in order of importance, the top-5 genes to emerge were ATP6V1C2, MYC, LMF1, HSPA4L, and PLAGL2.

Our findings are consistent with other published molecular signatures from previous studies (Zumwalt et al., 2016; He et al., 2018; Liu et al., 2019). Zumwalt et al. (Zumwalt et al., 2016) showed that ATP6V1C2 expression successfully

distinguished between cancerous and non-cancerous samples in CRC. He et al. (He et al., 2018) reported that the expression of c-Myc, which was one of the three related human genes encoded under MYC genes family, was observed in many human cancers and was elevated in up to $70 - 80\%$ in CRC. Liu et al. (Liu et al., 2019) [66] identified ten lncRNAs related to crucial outcomes in CRC, and one of these was LMF1. Zhang et al. (Zhang et al., 2018) obtained 34 genes using minimal redundancy maximal relevance (mRMR) and incremental feature selection (IFS) methods. They found that the HSPA4L gene was the most highly expressed in CRC patients with chromosomal instability (CIN) mechanism. Zheng et al. (Zheng et al., 2010) reported that the PLAGL2 gene was vital in increasing the effect on glioblastoma and colorectal cancer. Su et al. (Su et al., 2018) reported that PLAGL2 served as an oncogenic function in multiple human malignancies, including colorectal cancer (CRC).

This study was limited by the available number of homogeneous RNASeq and microarray datasets. Only one matched-pair set of 54 CRC samples was analyzed. Future studies should extend the approach to more than one cancer type and multiple datasets. However, the number of samples in each dataset ($n = 54$) ensured that the training and validation sets were large enough for the magnitude and statistical significance of the classification accuracies.

2.6 Conclusion

In summary, data integration by taking the intersection of the individual gene-lists from the two data types improved the classification accuracy of CRC. However, laboratory experiments should be conducted on this 23-gene signature to further assess its clinical significance in CRC research. NBLDA method was the best performer on the RNASeq data. Results suggest that the SVM method was the most suitable classifier for CRC across the two data types and had high accuracy before and after the integration. Future studies should determine the effectiveness of

integration in cancer survival analysis and the application on unbalanced data (where the classes are of different sizes) as well as on data with multiple classes.

Chapter 3

Paper II: A Stacking Ensemble Deep Learning Approach to Cancer Type Classification Based on TCGA Data

This chapter addresses the following objective:

- Develop and apply stacking ensemble approach based on deep learning methods to predict different cancer types.

3.1 Abstract

Cancer tumor classification based on morphological characteristics alone has been shown to have serious limitations. Breast, lung, colorectal, thyroid, and ovarian are the most commonly diagnosed cancers among women. Precise classification of cancers into their types is considered a vital problem for cancer diagnosis and therapy. In this paper, we proposed a stacking ensemble deep learning model based on one-dimensional convolutional neural network (1D-CNN) to perform a multi-class classification on the five common cancers among women based on

RNASeq data. The RNASeq gene expression data was downloaded from Pan-Cancer Atlas using GDCquery function of the TCGAbiolinks package in the R software. We used least absolute shrinkage and selection operator (LASSO) as feature selection method. We compared the results of the new proposed model with and without LASSO with the results of the single 1D-CNN and machine learning methods which include support vector machines with radial basis function, linear, and polynomial kernels; artificial neural networks; k-nearest neighbors; bagging trees. The results show that the proposed model with and without LASSO has a better performance compared to other classifiers. Also, the results show that the machine learning methods (SVM-R, SVM-L, SVM-P, ANN, KNN, and bagging trees) with under-sampling have better performance than with over-sampling techniques. This is supported by the statistical significance test of accuracy where the p-values for differences between the SVM-R and SVM-P, SVM-R and ANN, SVM-R and KNN are found to be $p = 0.003$, $p \leq 0.001$, and $p \leq 0.001$, respectively. Also, SVM-L had a significant difference compared to ANN $p = 0.009$. Moreover, SVM-P and ANN, SVM-P and KNN are found to be significantly different with p-values $p \leq 0.001$ and $p \leq 0.001$, respectively. In addition, ANN and bagging trees, ANN and KNN were found to be significantly different with p-values $p \leq 0.001$ and $p = 0.004$, respectively. Thus, the proposed model can help in the early detection and diagnosis of cancer in women, and hence aid in designing early treatment strategies to improve survival.

3.2 Background

Recent global public health research shows an epidemiological paradigm shift from infectious to non-communicable diseases, the latter including different types of cancers. The incidence and prevalence of cancer are on the increase worldwide, both in the developing and developed countries (Olsen, 2015; Morhason-Bello et al., 2013). The global cancer statistics estimated about 19.3 million new cancer cases in 2020 alone, and close to 10 million deaths of 36 cancers in 185 countries (Sung et al.,

2021). Breast cancer (with estimated 2.3 million new cases) is the most common diagnosed among women, followed by lung, colorectal, thyroid, and ovarian cancers. Moreover, the most leading cause of death is the lung cancer (with estimated 1.8 million deaths). The cancer burden is expected to increase to 28.4 million cases by 2040 (Sung et al., 2021).

Cancer tumor classification based on morphological characteristics alone has serious limitations in differentiating among cancer tumors and may cause a strong bias in identifying the tumor by experts (Golub et al., 1999; Mohammed et al., 2018; Tan & Gilbert, 2003). Recently, RNASeq gene expression data (Datta & Nettleton, 2014; Rai et al., 2018) has emerged as the preferred technology for the simultaneous quantification of gene expression compared to the DNA microarray (Koch et al., 2018; Zhao et al., 2016). The classification of cancer using gene expression data from RNASeq technology provides the opportunity to discriminate healthy and diseased samples or among different types and subtypes of cancer more accurately (García-Díaz et al., 2020). RNASeq gene expression data have had a profound impact on disease diagnoses and prognoses through accurate disease classification, which has helped clinicians to choose the appropriate treatment plans for patients (Abusamra, 2013). There exists striking disparities in the global cancers among women (Sung et al., 2021; Torre et al., 2017). Correct classification of these cancers is among the essential strategies to inform clinical decisions and reduce morbidity and mortality from cancers among women.

Although the use of gene expression data from RNASeq technology has improved cancer classification, it has its own limitations due to it being characterized by small samples sizes, with each sample having a large number of genes (the curse of dimensionality) (Yang & Naiman, 2014; Lusa et al., 2010). In addition, the samples also contain several genes that are uninformative and degrade the classification performance (García-Díaz et al., 2020; Vanitha et al., 2015). As a way to mitigate this

problem, it has been suggested to first perform filtration and feature selection through methods such as the two-sample t-test at a given stringent significance threshold before going further with model building (Haury et al., 2011). This procedure ensures that only informative and sufficiently differentially expressed genes between the outcome classes are used in building the classifiers. This process of feature selection motivates the evaluation of methods for the classification of different cancer tumors and disease stages, to improve early detection and the design of targeted treatment strategies that may reduce mortality. The two-sample t-test as a method for feature selection is easy to use but comes with the problem of multiple testing that the user has to deal with. Other methods or approaches that are model based, such as regularized regression methods, have recently become popularly used.

There are many supervised and unsupervised machine learning as well as deep learning methods developed for cancer classification using gene expression data. Several studies reported a higher predictive performance of the machine learning methods on the multi-class cancer classification problem (García-Díaz et al., 2020; Castillo et al., 2019; Ramaswamy et al., 2001; Nawaz et al., 2018). These studies, however, differ in the methods used for feature (gene) selection. In particular, Castillo et al. (Castillo et al., 2019) used differential expression analysis and minimum-redundancy maximum-relevance method for feature selection in the microarray and RNASeq data. García-Díaz et al. (García-Díaz et al., 2020) applied a grouping genetic algorithm for feature selection in five different cancers using RNASeq data.

Ramaswamy et al. (Ramaswamy et al., 2001), on the other hand, used support vector machines (SVM) and a recursive feature elimination method to remove the uninformative genes. These studies concentrated on the application of machine learning methods on a multi-class classification problem. Several methods

developed by other authors for multi-class cancer classification are reported to have a higher predictive performance compared to existing methods (Piao et al., 2017). Lee et al. (Lee et al., 2019) proposed a new ensemble classifier called cancer predictor using an ensemble model (CPEM), for classification of over 31 different cancer tumors downloaded from TCGA repository. In addition, they assessed different input features such as mutation profiles, mutations rates, mutation spectra, and signature. Thereafter, they investigated different machine learning and feature selection models in order to find the best model which achieved 84% of accuracy using 10 folds cross-validation. Furthermore, they used the six most common cancers out of 31 types and the model achieved 94% of classification accuracy. However, some of the statistical methods achieved results that are better than machine learning algorithms. Tabares-Soto et al. (Tabares-Soto et al., 2020) compared machine learning and deep learning methods in classifying 11 different tumor classes using microarray gene expression data. They implemented eight supervised machine learning methods including KNN, support vector classifier (SVC), logistics regression (LR), linear discriminant analysis (LDA), naïve Bayesian classifier (NB), multi-layer perceptron (MLP), decision trees, and random forest (RF) as well as one unsupervised method such as k-means. In addition, they applied two deep neuronal networks (DNN) methods. Their results showed that the deep learning methods outperformed the other machine learning methods.

In this study, we propose a stacking ensemble deep learning model that uses five 1D-CNN as base models. The results of these models are combined using NN, which is used as a meta model to classify the most common types of cancers among women using RNASeq data. We compared the performance of our new proposed model when using the full list of genes as input with its performance when using a reduced selection of genes using LASSO. Also, we consider comparing the performance of our current proposed model with other machine learning methods since there are limited studies that compare the performance of deep learning and

machine learning methods to classify different types of cancer. LASSO is used as a feature selection technique, since it has been shown to improve prediction accuracy, especially when there is a small number of observations and a large number of features (Fonti & Belitser, 2017). Findings from this study might help in the early detection and accurate classification of these cancer types and contribute to efforts of finding therapies that may increase survival for women at risk.

3.3 Material and methods

In this paper, we downloaded the RNASeq gene expression data from Pan-Cancer Atlas (https://portal.gdc.cancer.gov/), using *R* statistical software version 3.6.3 via the *TCGAbiolinks* package (Chang et al., 2013; Colaprico et al., 2016; R Core Team, 2020). The data contains 2166 samples from the top five common cancers between women. We applied eight multi-class classification methods to find the best classifier that discriminates among five common cancers among women. The machine learning methods were implemented in the *R* software, while the deep learning method (1D-CNN) was implemented using *TensorFlow* with *Keras*.

3.3.1 Datasets

We used only five cancer tumors (normal cases were excluded) from RNASeq gene expression datasets. The cancer tumors were breast, colon adenocarcinoma, ovarian, lung adenocarcinoma, and thyroid cancer. The datasets were downloaded from Pan-Cancer Atlas using *GDCquery* function of the *TCGAbiolinks* package in *R* (Colaprico et al., 2016). *GDCquery* function has many parameters, to define the function known by the following names: project, legacy, data.category, data.type, platform, file.type, experimental.strategy, and sample.type. The project parameter indicates a list of the data that should be downloaded. In our case, we passed the five project codes corresponding to our five types of cancer, which are TCGA-BRCA, TCGA-COAD, TCGA-OV, TCGA-LUAD, and TCGA-THCA. We set the legacy to "true", which helps the query to search only in the legacy repository

for the unmodified stored data in the TCGA data portal.

"Gene expression" and "Gene expression quantification" are passed to the data.category and data.type arguments, respectively, to filter the data files to be downloaded. The platform "Illumina HiSeq" was used to download the data. We used "results" for file.type argument to filter the legacy database, and "RNA-Seq" was chosen as experimental.strategy argument to produce the expression profiles. Moreover, we selected the tumor samples to be downloaded using the "Primary solid Tumor" value as sample.type argument. The downloaded data in a matrix form included five types of cancer, where the columns represent the samples and the rows containing the genes, i.e. features (equivalently covariates). The datasets were combined to give 2166 tumor samples obtained from all the five cancers, with $19,947$ common genes. Due to the curse of high-dimensionality, we performed filtration and feature selection to reduce the high number of genes in order to exclude irrelevant and noisy ones that could affect the performance of the methods. Thus, we applied normalization, transformation, and filtration steps to the data to select the informative genes that potentially could contribute positively to the classification accuracy. Table 3.1 below shows a summary of the downloaded data including the training and testing fractions for each cancer tumor.

Table 3.1: Number of samples in each class used in the classification.

Cancer tumor	**Number of samples (%)**	**Training (**$\approx 70\%$**)**	**Testing (**$\approx 30\%$**)**
Breast (BRCA)	1082 (50)	753	329
Colon adenocarcinoma (COAD)	135 (6)	99	36
Lung adenocarcinoma (LUAD)	275 (13)	189	86
Ovarian (OV)	304 (14)	217	87
Thyroid (THCA)	370 (17)	259	111
Total	2166	1517	649

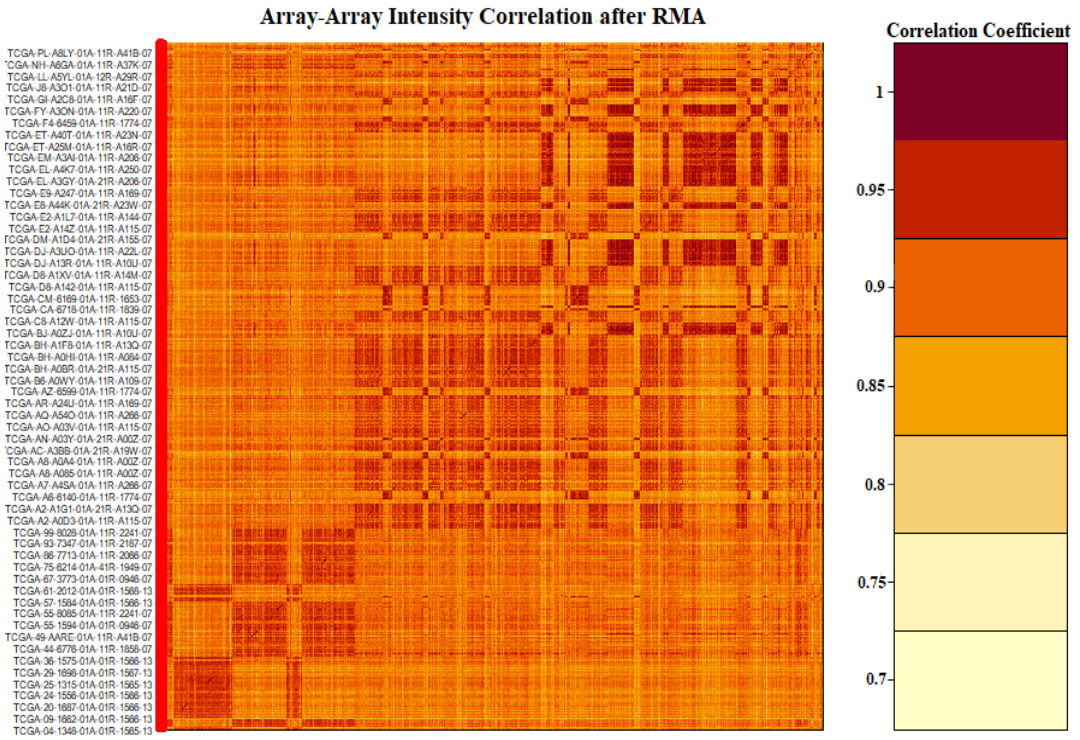


Figure 3.1: Array-array intensity correlation (AAIC) matrix defines the Pearson correlation coefficients among the samples.

3.3.2 Data pre-processing

We used TCGAanalyze_Preprocessing function in *TCGAbiolinks* package (Colaprico et al., 2016), which utilizes an array-array intensity correlation (AAIC) approach to obtain a $N \times N$ square symmetric matrix of Spearman correlations among the samples. The AAIC enabled us to find samples with low correlation considered as possible outliers (Fig 3.1). After that, we performed gene normalization through TCGAanalyze_Normalization function, which calls the sub-routines newSeqExpressionSet, withinLaneNormalization, betweenLaneNormalization, and counts from *EDASeq* package to adjust the GC-content effect or other gene level effects, distributional differences between lanes, and global-scaling and full-quantile normalization (Bullard et al., 2010). TCGAanalyze_Filtering was used for filtering out the irrelevant genes and returned the genes with the mean intensity across the samples higher than 0.25, which was the threshold defined quantile

mean. After applying this process, we found $14,899$ genes to be informative meaning 5048 genes were rendered irrelevant. For further reduction and precise differential gene expression analysis, we used *DESeq* package in *R* (Anders & Huber, 2010; Love et al., 2019; Dündar et al., 2015). *DESeq* analyses the gene expression based on the negative binomial distribution and a shrinkage estimator for the distribution's variance. After using *DESeq* package, $12,649$ genes out of the $14,899$ post initial filtering were found to be differentially expressed meaning a further 2250 genes were removed.

3.3.3 Feature selection using LASSO regression

The RNASeq gene expression data after preprocessing had $12,649$ dimensions or features, which was still huge given that the number of samples was 2166. Therefore, LASSO regression was used to decrease the number of genes or features that enabled us to effectively analyze the data. LASSO is a method that performs regularization and feature selection through a shrinkage (regularization) process. LASSO penalizes the regression coefficients with L1-norm whereby some coefficients are shrunk to zero. After that, the coefficients of the regression variables having significantly non-zero values are selected and used in the model (Fonti & Belitser, 2017).

In the case of the multinomial response with $K > 2$ levels, assume that $\ell(g_i) = Pr(C = c_i \,|\, g_i)$, where $c_i \in 1, 2, 3, \ldots, K$ is the i^{th} response. The log-likelihood of the multinomial model under LASSO model can be written in a generalized form as (Friedman et al., 2010)

$$\underset{\{\beta_{0\ell},\beta_\ell\}ss_1^K \in \mathbb{R}^{K(p+1)}}{max} \left[\frac{1}{N} \sum_{i=1}^{N} log\, p_{c_i}(g_i) - \lambda \sum_{\ell=1}^{K} P_\alpha(\beta_\ell) \right], \tag{3.1}$$

which can be maximized as a penalized log-likelihood. The outcomes in the data can be denoted in the form of a matrix $\boldsymbol{Y}$ of dimension $N \times K$, with elements $y_{i\ell} = I(c_i = \ell)$. Thus, the terms in the regularized log-likelihood in Eq. 3.1 can be written in more

explicit form

$$\ell(\{\beta_{0\ell}, \beta_\ell\}_1^K) = \frac{1}{N}\sum_{i=1}^{N}\left[\sum_{\ell=1}^{K} y_{i\ell}(\beta_{0\ell} + g_i^T\beta_\ell) - log\left(\sum_{\ell=1}^{K} e^{\beta_{0\ell}+g_i^T\beta_\ell}\right)\right], \tag{3.2}$$

$$P_\alpha(\beta) = (1-\alpha)\frac{1}{2}\|\beta\|_{L_2}^2 + \alpha\|\beta\|_{L_1} \tag{3.3}$$

$$= \sum_{j=1}^{p}\left[\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j|\right], \tag{3.4}$$

P_α is the penalty part, where g_i is the gene expression levels for sample i, β_ℓ is the vector of the regression coefficients, $y_{i\ell}$ is the class response value in sample i. When $\alpha = 0$ in Eq. 3.3 we obtain the ridge regression penalty, whereas $\alpha = 1$ leads to LASSO regression penalty. We chose LASSO regression because it uses the sum of the absolute values of the model parameters, restricted to be less than a fixed value as the penalty. LASSO, with tenfold cross-validation returned 173 genes (Supplementary File 5.4). These genes were obtained when lambda (λ) value gave a minimal deviance associated with the response variable, and so were used for the classification. The cross-validated multinomial deviance is a function of $log(\lambda)$, and when $log(\lambda)$ is equal to -1, it is an indication that λ and multinomial deviance are both big. As λ decreases and becomes very small, the multinomial deviance also becomes small and almost flat, indicating that the attained model is a good fit. There are many advantages of the LASSO method, which include removing those variables with zero coefficients that lead to reduced variance without an intrinsic increase in bias. The method also minimizes over-fitting by excluding irrelevant variables that are not related to the outcome variable. The LASSO method naturally also deals with the multiple testing problem, by penalizing irrelevant features, whose contribution is shrunk to zero. This leads to an improved classification and prediction accuracy (Fonti & Belitser, 2017; Pereira et al., 2016). In our case, LASSO was implemented using *glmnet* package in *R* (Hastie et al., 2016).

3.3.4 Data partitioning

We used tenfold cross-validation to evaluate the different prediction methods using 70% of the dataset. In the tenfold cross-validation, the dataset is divided into ten parts, where one part is removed to represent the validation set, and the remaining nine parts combined to represent the training set. Thus, this process is repeated ten times by removing one part each time to have a different part of the data for validation (Hu et al., 2006). We left aside 30% of the entire dataset, which served as an independent testing set for the final evaluation.

3.3.5 The classification models

We performed classification on the different cancers as a multi-classification problem using gene expression levels as covariates. Eight classification methods were used: the new proposed stacking ensemble deep learning model; one-dimensional convolutional neural network (1D-CNN); support vector machines (SVM) with radial basis function, linear, and polynomial kernels; artificial neural networks (ANN); K-nearest neighbors (kNN); and bagging trees.

Support vector machines (SVM) (Boser et al., 1992), is a well-known machine learning method that has been used widely in many fields, including gene expression data analysis (Brown et al., 1999; Chu & Wang, 2003). SVM aims to find an optimal hyperplane that separates the data into two different classes for the binary classification problem, determined by a subset of samples known as support vectors (Munoz et al., 2003). SVM can handle non-linearly separable problems by transforming the data using mapping kernel functions. These functions include radial basis, polynomial, and linear functions (Stephens & Diesing, 2014). The SVM is implemented using *kernlab* package in *R* statistical software (Karatzoglou et al., 2019). Suppose we have n samples and p genes. Furthermore, assume samples belong to two linearly separable classes represented by $+1$ or -1, and suppose $\mathbf{g_i}$ is the features vector. Then we let, $(\mathbf{g}_i, \mathbf{y}_i) \in G \times Y, i = 1, 2, 3, \ldots, n$, where

$y_i \in +1, -1$ is the target variable dichotomy in the p dimensional space. The aim is to classify the sample into one of the two classes and by extension find an SVM classifier that generalizes to a multi-class problem. There are many hyperplanes that discriminate the two classes, but the goal is achieved by finding an optimal separating hyperplane that lies furthest from the both classes. The separating hyperplane can be defined by

$$\mathbf{w} * \mathbf{g} + b = 0, \tag{3.5}$$

where w is the weight vector, b is the bias, and $\frac{|b|}{\|\mathbf{w}\|}$ is the perpendicular distance to the hyperplane. We can rescale the w and b such that the following equation determines the point in each class that is nearest to the hyperplane defined by the equation

$$\mathbf{w} * \mathbf{g} + b = 1. \tag{3.6}$$

Therefore, a separating hyperplane for the two classes should follow

$$\mathbf{w} * \mathbf{g} + b = +1 \quad \text{when } y_i = +1. \tag{3.7}$$

$$\mathbf{w} * \mathbf{g} + b = -1 \quad \text{when } y_i = -1. \tag{3.8}$$

After the rescaling, the distance from the nearest point in each class to the hyperplane becomes $\frac{1}{|\mathbf{w}|}$. Consequently, the distance between the two classes is $\frac{2}{\|\mathbf{w}\|}$, which is called the margin. The solution of the following optimization problem is obtained by maximizing the margin:

$$\underset{w,b}{min} \frac{1}{2}\|\mathbf{w}\|^2 \text{ subject to } y_i(\mathbf{w} * \mathbf{g} + b) \geq 1, \ i = 1, 2, 3, \ldots, n. \tag{3.9}$$

For the multi-class problem there are many types of extensions that can be used such as one-vs-one, one-vsall (one-vs-rest), decision directed acyclic graph based approach, multi-class objective function, and errorcorrecting output code based

approach. These approaches use the same binary classification principle, where the multi-class problem is decomposed into multiple binary problems. In the one-vs-one multi-class classification problem the SVM classifier produces all possible pairs of binary classifications. Suppose we have k classes where $k > 2$, then, $\frac{k(k-1)}{2}$ binary classifiers are produced in the training step of the algorithm. Consequently, a sample in the test dataset is assigned the class label that is voted the most by the $\binom{k}{2}$ binary classifiers from the trained one-vs-one SVM. In our case we use the one-vs-one multi-class classifier.

Artificial neural networks (ANN) is a computational method constructed from many layers, each layer consisting of nodes called neurons (Dwivedi, 2018). The data flows from the input layer to the output layer through the hidden layers (Lek & Park, 2008). The nodes between the input through the hidden layers to the output layers are connected by appropriately defined weights or weight functions. The number of input and output layers depends on the number of covariates in the dataset as well as a number of classes in the outcome variable (Lek & Park, 2008). The inputs are weighted by multiplying every one of them by a weight which is a measure of its contribution. Therefore, the hidden layer receives the weighted inputs and produce outputs using an activation function(s) (Stephens & Diesing, 2014; Dwivedi, 2018). ANN can be implemented using the *R* package *nnet* (Ripley et al., 2016).

Specifically suppose we have gene expression data with p genes. The input layer receives the p genes and multiplies them by weights as follow

$$b_i = \sum_{j=0}^{P} w_{ij}^{(1)} g_i \quad i = 1, 2, 3, \ldots, n, \tag{3.10}$$

where g is a vector of input features and $g_0 = 1$ is a constant input feature with weight w_{i0}. The b_i are called activations, and the parameters $w_{ij}^{(1)}$ are the weights. The subscripts (1) refer to the first layer of the network. Then the activations are

transformed by a nonlinear activation function f, usually a sigmoid function as given in the following equation

$$z_i = f(b_i) = \frac{1}{1 + exp(-b_i)}. \tag{3.11}$$

In the second layer, the outputs of the hidden units are linearly combined to give the activations

$$h_k = \sum_{i=0}^{n} w_{ik}^{(2)} z_i \quad k = 1, 2, 3, \ldots, K, \tag{3.12}$$

where the $w_{ik}^{(2)}$ are the weight parameters for the transformation in the second layer of the neural network. The outputs are transformed using an activation function such as the sigmoid function

$$y_k = f(h_k) = \frac{1}{1 + exp(-h_k)}. \tag{3.13}$$

K-nearest neighbors (kNN) is a non-parametric method used for classification and regression (Yao & Ruzzo, 2006). The idea behind kNN lies in finding the most nearest neighbors of the new sample, and this is based on the similarity and distance metric (Cunningham & Delany, 2020). In kNN, *k*-neighbors determine the class of a new instance; therefore, the new sample is assigned the class that is most likely among the *k*-neighbors (Stephens & Diesing, 2014; Dwivedi, 2018). In general, kNN has two phases; the first is finding the nearest neighbors, and the second is assigning the class of a new sample using those neighbors by the majority vote rule. kNN is implemented using *R* package *caret* (Kuhn et al., 2020).

Suppose we have two samples s_1, s_2 each with p genes. Since kNN uses the Euclidean distance measure to find the closest sample for a new sample, the distance between the two samples can be calculated as

$$dist(s_1, s_2) = \sqrt{\sum_{j=1}^{P} (g_{1j} - g_{2j})^2}. \tag{3.14}$$

A new sample is allocated the class that most of its neighbors fall, that is, model class of its neighbors.

Bagging trees or bootstrap aggregation method is appealing because its ability to reduce the variance associated with a prediction and hence, improve the prediction accuracy (Sutton, 2005). The method splits the data into many bootstrap samples, thereafter, train the model for each bootstrap. Then, the overall prediction obtained by averaging and voting for regression and classification, respectively.

Convolution Neural Networks (CNNs) are deep learning architectures that have multi-layers between the input and output and are designed for image analysis and classification (Bengio, 2009; Schmidhuber, 2015; Elbashir et al., 2019). Deep learning is applied successfully in many areas including medical image analysis, computer vision, drug design, and bioinformatics and yield performance that sometimes surpass expert personals' performance (Ciregan et al., 2012). CNNs are a regularized version of fully connected networks (multilayer perceptrons), in which each neuron in one layer is connected to all the neurons in the layer that follows it. The connectivity between the neurons is inspired by the biological process and resembles the arrangement of the animal visual cortex. In contrast to other image classification and analysis algorithms, CNNs use little pre-processing by learning the filters that capture temporal and special dependencies in an image instead of hand-engineering them. A sequence of stacked layers (convolutional layer, pooling layer, and fully-connected layer) makes the architecture of CNNs and in each layer, a differentiable function is used to transform one volume of activations to the layer that follows it. The major building blocks in CNNs are the convolutional layers, which apply filters on an input image to create a feature map. To get a good classification performance, CNNs normally decrease the features of the image into

an easier processed arrangement without dropping essential features. The pooling layers use max pooling or average pooling to reduce the dimension of the image's features. The fully connected layer is an important component in the CNNs architecture that derives the final classification results.

The input to the CNNs is a tensor of order 3 that represents an image having m rows and n columns with 3 color channels (RGB). The tensor encodes the pixel intensities of the image and produces the input features that go through the convolutional, pooling, and the fully connected layers sequentially. In the convolutional layer, a filter of size f by f and stride $= s$ are applied and the result is $3 \times (m - f + 1) \times (n - f + 1)$ hidden feature neurons if a stride of 1 is used and the pooling layer result will be $3 \times (m - r + 1)/2 \times (n - r + 1)/2$ hidden features neurons when applied to 2×2 regions. The convolution operation generates the features map by multiplying the element of the input array by the element of the filter element wise and summing up the result to generate on pixel of the features map. Sliding the filter across the matrix and repeating the multiplication and summing up operations will generate the rest of features map pixels. The mathematical equation of this convolution operation is given as follows

$$O(i,j) = \sum_{k=1}^{f} \left(\sum_{l=1}^{f} \mathit{input}(i+k-1, j+l-1)\, \mathit{kernel}(k,l) \right), \tag{3.15}$$

where $i = 1, 2, \ldots, m - f + 1$, $j = 1, 2, \ldots, n - f + 1$.

1D-CNN is a simple CNN architecture that has only one convolutional layer. The simple design of this model leads to reduced number of parameters that can be adjusted during the training process therefore, it is highly needed in the genomic studies where it is difficult to collect large data to train a deep learning model that has very large number of parameters (Mostavi et al., 2020). The one dimensional that we used in this study was constructed by Mostavi et al. (Mostavi et al., 2020) for predicting cancer tumor based on gene expression data. The architecture of the

model when using LASSO as a feature selection technique is shown in Fig. 3.2.

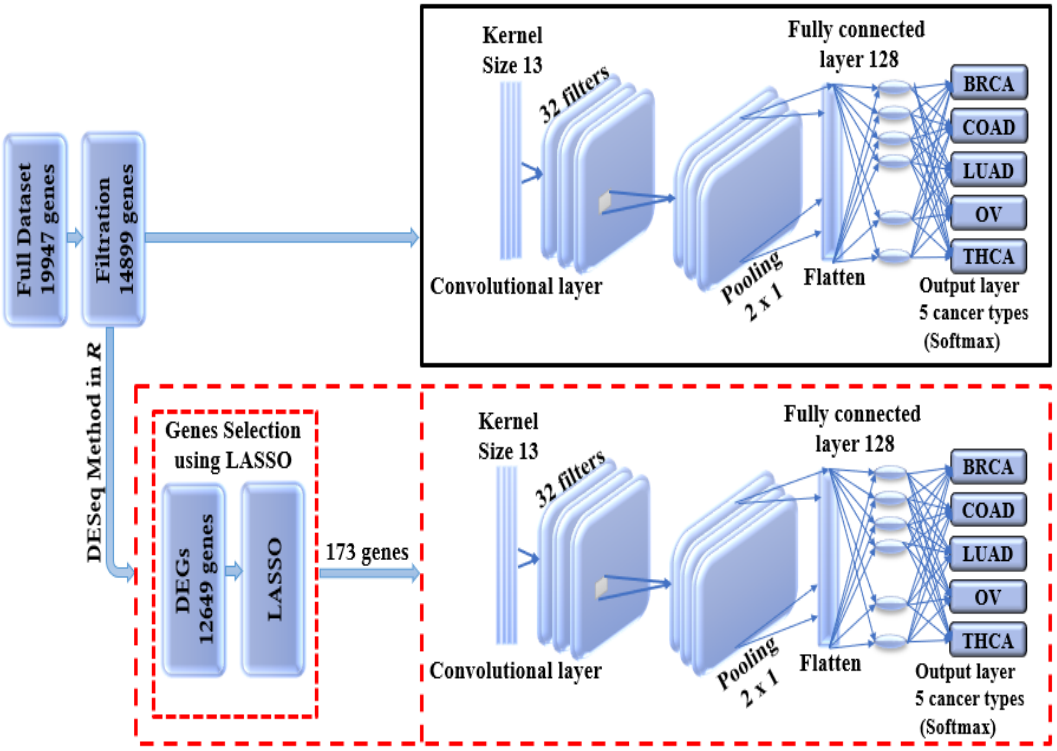


Figure 3.2: Illustrates the architecture of the 1D-CNN model. The upper panel presents the 1D-CNN without LASSO, while the lower panel shows the usage of LASSO as s feature selection technique for the 1D-CNN where it gives an input vector with 173 genes.

3.3.6 Regularization with early stopping

We applied 1D-CNN with early stopping regularization to avoid over-fitting. The over-fitting is usually caused by training the model too much, making it pick up the noise as an essential part of the data instead of relying only on the training data. Such noise is normally unique to each training data. It can lead to high variance in the model estimates. On the other hand, too little training can result in under-fitting or high bias. Therefore, the variance and the bias have a negative relationship meaning that if the bias increases for fixed mean square error, then the variance will decrease and vice versa and that is known as the bias-variance tradeoff (Friedman et al., 2001; Yang et al., 2020). To avoid over-fitting, we can use a model with fewer parameters or obtaining more data. A model with fewer parameters can cause high

bias. Since obtaining more data is not easy in the medical field, then a model with fewer parameters seems to be the alternative, but modern approaches in deep learning repeatedly show the benefits of using models with a large number of parameters (Krizhevsky et al., 2012; Simonyan & Zisserman, 2014). Therefore, finding a way of adjusting the variance by minimizing noisy data can help solve the over-fitting problem. Since too much training can result in over-fitting, whereas too little training can result in under-fitting then the model can be regularized using the early stopping mechanism. We can implement the early stopping mechanism in the training procedure to make the architectures better fit the training data with each epoch and determining the number of epochs that can be run before the pre-trained model begin to overfit.

3.3.7 Stacking ensemble

Ensemble learning is the process of improving classifiers performance by combining the contribution of the trained sub-models to solve same classification problem (Mohammed et al., 2018). Overall, each base learner votes and the final prediction is gained by the meta-learner, which is a model that learn to correct the prediction of the base-learners. Therefore, the ensemble approach results in prediction accuracy that is better than the single learners. Generalizability of an ensemble usually reduces the variance in the prediction, and thus ensure the most stable and best possible prediction is made. The meta model takes the output of the sub-models (baselearners) as input and then learns to merge the input prediction to make the final prediction which is better than each of the base-classifiers. Fig. 3.3 shows our proposed stacking ensemble deep learning model.

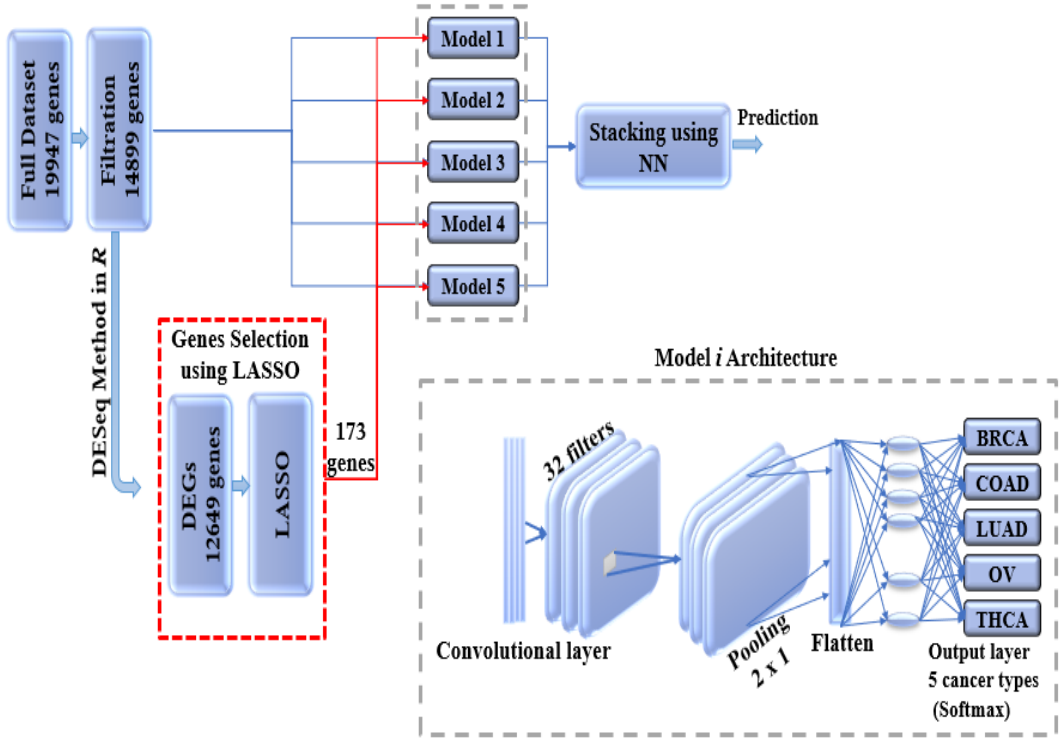


Figure 3.3: Stacking ensemble deep learning model architecture in which five 1D-CNN models are used as base models and the results of these models are combined using NN, which is used as a meta model. The NN has one hidden layer and an output layer that is activated using softmax function.

3.3.8 Performance evaluation

We used different performance metrics to evaluate the performance of the classification methods. These metrics are namely accuracy, kappa, specificity, sensitivity, the area under the curve (AUC), precision, F-measure, and ROC curve. The accuracy measures the percentage of correctly classified cases but is not sufficient for measuring the performance of the classifier, especially if we have unbalanced data (which is the case with cancer data that we are dealing with). Sensitivity measures the percentage of the cases that are correctly classified as having cancer among those samples that are truly cancerous. Therefore, it measures the fraction of the correctly predicted cancer cases. Specificity measures the percentage of cases that do not have cancer, which are correctly identified to be so. In other words, it measures the true negative rate. Precision is the percentage of

cases among those classified as positive that are truly positive, i.e., having cancer, and sometimes this measure is called the positive predicted value. F-measure is a measure that balances between precision and sensitivity.

We also compared the predictive performance of the methods using the receiver operating characteristic (ROC) curve plots. These figures were plotted using *MultiROC* package in *R* (Wei et al., 2018). *MultiROC* calculates and visualizes ROC curve for multi-class using *micro-averaging* and *macro-averaging* approaches. *Micro-averaging* ROC-AUC converts the multi-class classification into binary classification by stacking all groups together. *Macro-averaging* ROC-AUC uses one versus the rest approach by averaging all group's results and linear interpolation used between the points of the ROC. Confidence intervals for kappa statistics were computed using *vcd* package.

3.3.9 Methods to adjust for class imbalances

Imbalanced class sizes may lead to poor predictive performance particularly for the classes with small samples (Table 3.1). In order to handle the class imbalance and hence improve the models' performances we used the synthetic minority over-sampling technique (SMOTE) and under-sampling (DOWN) methods. SMOTE has been used widely in various fields such as bioinformatics for addressing the class imbalance in the outcome (Xiao et al., 2011; Batuwita & Palade, 2009). SMOTE is a data augmentation method that add new data to the minority class that are synthesized from the existing data instead of duplicating the data, because the duplication will not provide any new information to the model. SMOTE works by first selecting randomly a class instance a from the minority class then it chooses randomly one of the k nearest neighbors b to create the synthetic instances as a convex combination of a and b and finally, it forms a line segment in the feature space by connecting a and b.

We synthesized the minority class from existing samples by selecting randomly the closest k minority nearest neighbors to balance the class (Chawla et al., 2002; Chawla, 2009; Johnson & Khoshgoftaar, 2019). This statistical technique increases and generates the samples to reach the highest majority class and it makes the samples more general. SMOTE is implemented using *caret* package in *R* by adjusting the sampling method in the train control parameter to be 'SMOTE'.

Under-sampling technique (DOWN) tends to produce a new balanced subset of the original dataset by randomly removing instances usually from the majority class observations (Galar et al., 2011; Rok & Lara, 2013). DOWN is implemented using *caret* package in *R* by adjusting the sampling method in the train control parameter to be 'DOWN'.

3.3.10 Statistical significance test

There are many different techniques that can used for comparing the accuracies of the machine learning models. In this work, we used the *resamples* method in *R* to analyze and visualize the estimated performance of the models. We used the *summary* function to compute summary statistics across each model/metric combination. *Diff* function in *R* is used to estimate the differences between the methods. The *diff* function performs a pairwise comparisons to compute the differences between pairs of consecutive elements using Bonferroni correction as an adjustment method. Bonferroni test is a type of multiple testing method used in statistical analysis to reduce the instance of a false positive and prevent the data from appearing incorrectly to be statistically significant (Trawiński et al., 2012; Wang et al., 2017).

3.4 Results

We found that the performance of the machine learning methods when LASSO as feature selection technique used is by far better than when it is not used. The

performance of the methods in terms of overall statistics are summarized in Table 3.2 based on the under-sampling technique. Table 3.3 shows the results of methods in terms of per-class statistics for under-sampling technique. The receiver operating characteristic (ROC) curve plots comparing the machine learning classification methods in this study are shown in, Figs. 3.4, 3.5, 3.6, 3.7, 3.8, and 3.9 based on under-sampling method. The predictive performance of the under-sampling technique outperformed the over-sampling technique. Results for the over-sampling technique are available in the Supplementary File 5.4.

3.4.1 The overall predictive performance of the machine learning methods based on the under-sampling technique

The accuracy, precision, sensitivity, and F1-Score performance measures for the overall multi-class classification problem based on the under-sampling technique (DOWN) are presented in Table 3.2. These results show that bagging trees method achieved the best performance measure compared to the other methods where it yields an accuracy, sensitivity, AUC, and F1-score of 99.2%, 99.4%, 99.54%, and 99.5%, respectively. However, SVM-P and bagging trees have the same precision, and they have a close results in the other performance measures. Consequently, ANN method obtained the worst performance with an accuracy of 80.7%.

Table 3.2: The overall predictive performance of the machine learning methods based on under-sampling.

Methods	Performance Measures					
	ACC (95% CI)	Kappa (95% CI)	F1-Score	Precision	Sensitivity	AUC
SVM-R	95.84 (94.00, 97.24)	93.81 (91.55, 96.07)	98.64	99.39	97.90	98.04
SVM-L	96.76 (95.10, 97.99)	95.14 (92.74, 97.18)	97.48	100.0	95.08	98.56
SVM-P	98.92 (97.79, 99.57)	98.40 (97.89, 99.74)	99.24	99.69	98.79	99.50
ANN	80.74 (77.49, 83.71)	72.15 (70.39, 79.59)	87.46	84.80	90.29	83.84
kNN	93.07 (90.83, 94.90)	89.97 (87.18, 92.75)	95.91	92.70	99.34	94.94
Bagging trees	99.20 (98.21, 99.75)	98.86 (97.86, 99.85)	99.54	99.69	99.39	99.54

Note: SVM-R, Support Vector Machine with Radial-basis function (RBF) kernel; SVM-L, Support Vector Machine with Linear Kernel; SVM-P, Support Vector Machine with Polynomial Kernel; ANN, Artificial Neural Networks; kNN, K-nearest Neighbors; Bagging trees; ACC, Accuracy; CI, Confidence Interval; Kappa, Kappa Statistics; AUC, Area Under the Curve.

3.4.2 Predictive performance of the machine learning methods per cancer tumor based on the under-sampling

The accuracy, precision, sensitivity, and F1-Score performance measures based on perclass statistics using the under-sampling technique method (DOWN) are presented in Table 3.3. Bagging trees outperforms the other methods in classifying most of the five cancer tumors in most of the performance measures, followed by SVM-P method. While the ANN shows the lowest performance measures. These results were confirmed using the ROC curves which are depicted in Figs. 3.4, 3.5, 3.6, 3.7, 3.8, and 3.9. Bagging trees was able to highly correctly classify the ovarian cancer with 100% in terms of accuracy, sensitivity, specificity, F1-Score, and precision. While SVM-L and SVM-P can sensitively classify the thyroid cancer with a 100% of accuracy, sensitivity, specificity, F1-Score, and precision. Also, SVM-R shows performance that is close to SVM-L and SVM-P to classify the thyroid cancer.

Table 3.3: Predictive performance of the machine learning methods per-class statistics based on undersampling.

Performance		Methods					
Measures	**Class**	**SVM-R**	**SVM-L**	**SVM-P**	**ANN**	**kNN**	**Bagging trees**
	BRCA	98.6	97.3	99.2	87.7	96.0	99.5
	COAD	95.8	98.6	98.6	90.2	94.7	98.5
Accuracy	LUAD	97.7	99.6	98.0	82.8	90.6	98.7
	OV	90.7	88.5	98.9	93.4	98.5	100
	THCA	97.8	100	100	82.5	99.1	99.6
	BRCA	99.4	100	99.7	84.8	92.7	99.7
	COAD	91.7	97.2	97.2	86.1	94.4	97.2
Sensitivity	LUAD	98.8	100	96.5	68.6	81.4	97.7
	OV	81.6	77.0	97.7	92.0	98.9	100
	THCA	95.5	100	100	67.6	98.2	99.1
	BRCA	97.8	94.7	98.8	90.6	99.4	99.4
	COAD	100	100	100	94.3	94.9	99.8
Specificity	LUAD	96.6	99.3	99.5	97.0	99.8	99.6
	OV	99.8	100	100	94.8	98.0	100
	THCA	100	100	100	97.4	100	100
	BRCA	98.6	97.5	99.2	87.5	95.9	99.5
	COAD	95.7	98.6	98.6	60.8	67.3	97.2
F1-Score	LUAD	89.5	97.7	96.5	72.8	89.2	97.7
	OV	89.3	87.0	98.8	81.6	93.5	100
	THCA	97.7	100	100	75.0	99.1	99.6
	BRCA	97.9	95.1	98.8	90.3	99.4	99.4
	COAD	100	100	100	47.0	52.3	97.2
Precision	LUAD	81.7	95.6	96.5	77.6	98.6	97.7
	OV	98.6	100	100	73.4	88.7	100
	THCA	100	100	100	84.3	100	100

Note: SVM-R, Support Vector Machine with Radial-basis function (RBF) kernel; SVM-L, Support Vector Machine with Linear Kernel; SVM-P, Support Vector Machine with Polynomial Kernel; ANN, Artificial Neural Networks; kNN, K-nearest Neighbors.

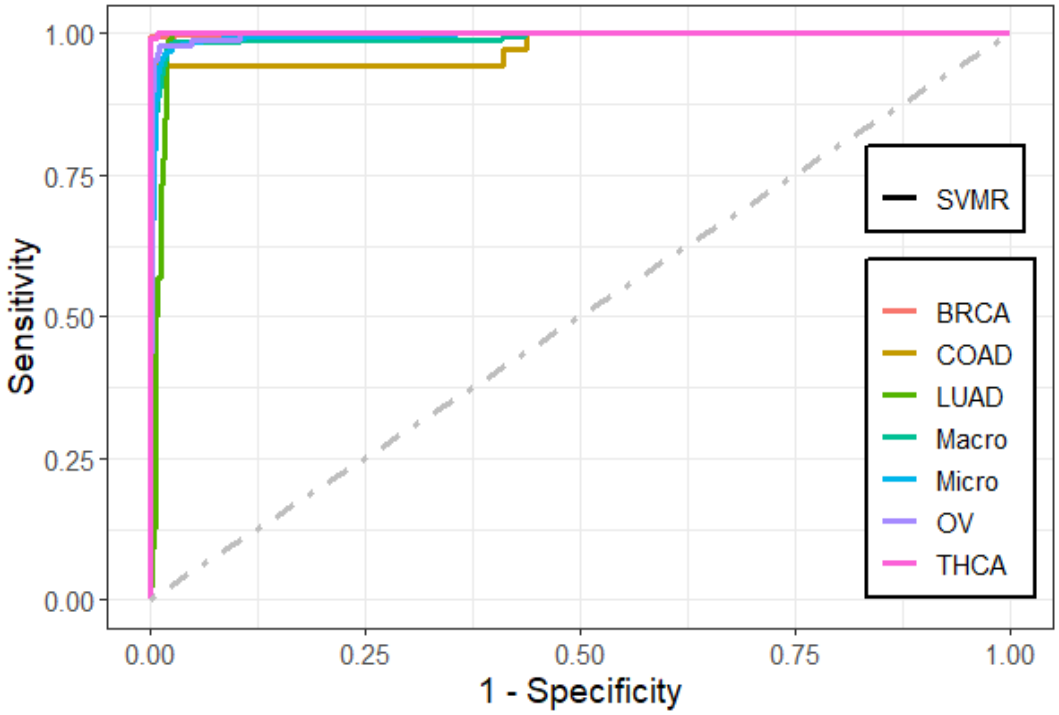


Figure 3.4: Multi-class ROC curves visualization for the SVMR model based on under-sampling technique.

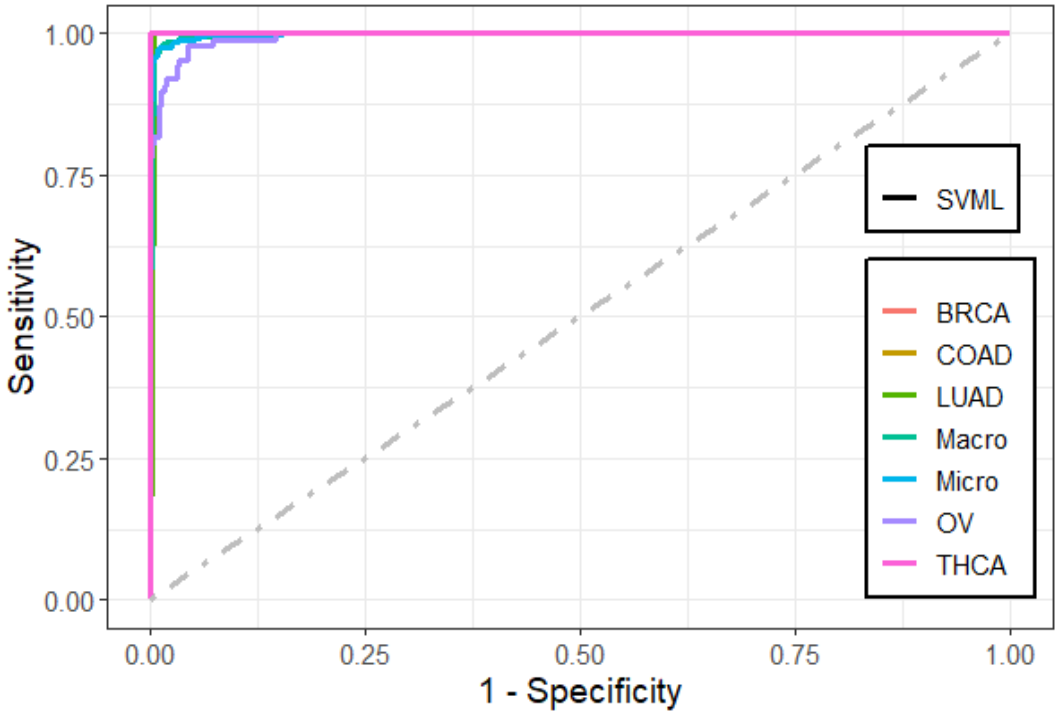


Figure 3.5: Multi-class ROC curves visualization for the SVML model based on under-sampling technique.

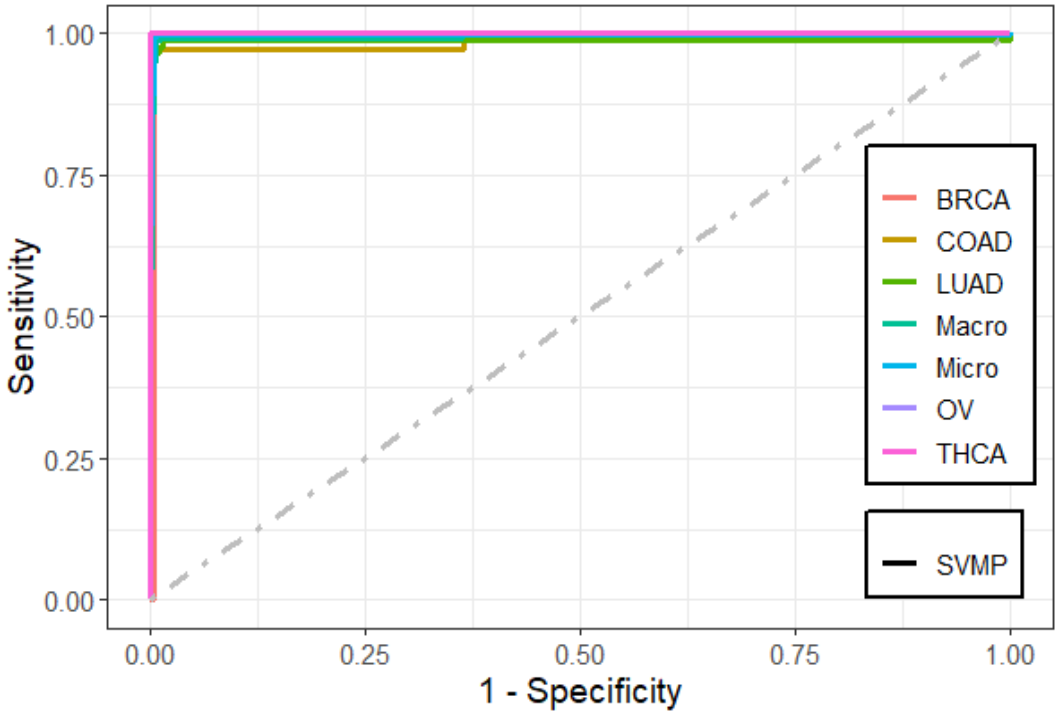


Figure 3.6: Multi-class ROC curves visualization for the SVMP model based on under-sampling technique.

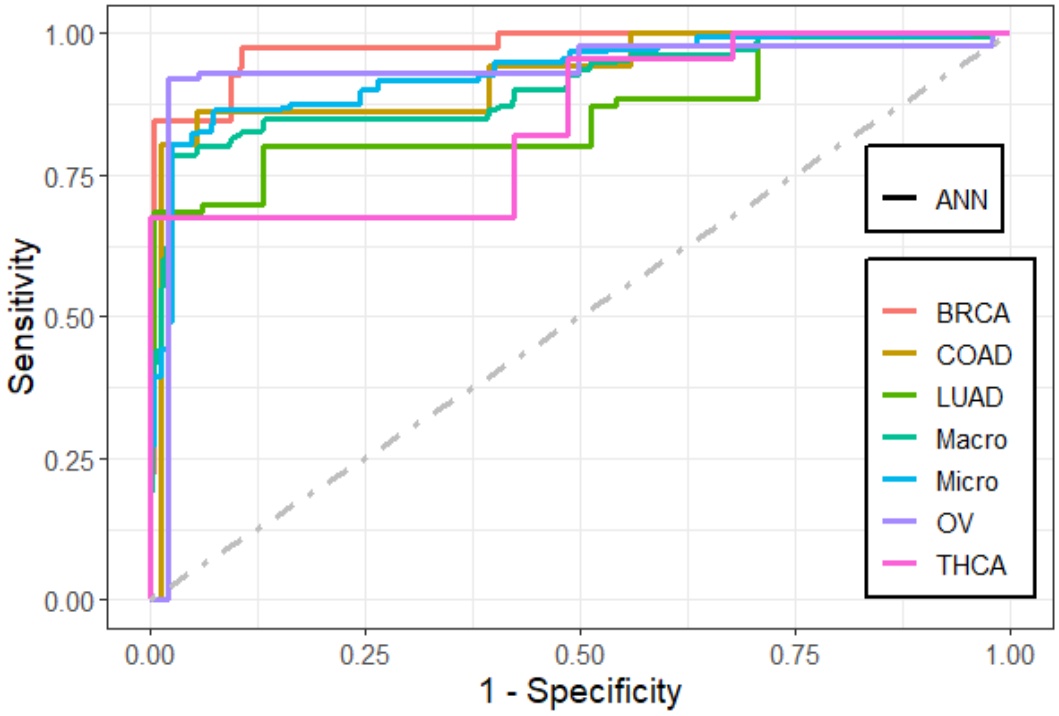


Figure 3.7: Multi-class ROC curves visualization for the ANN model based on under-sampling technique.

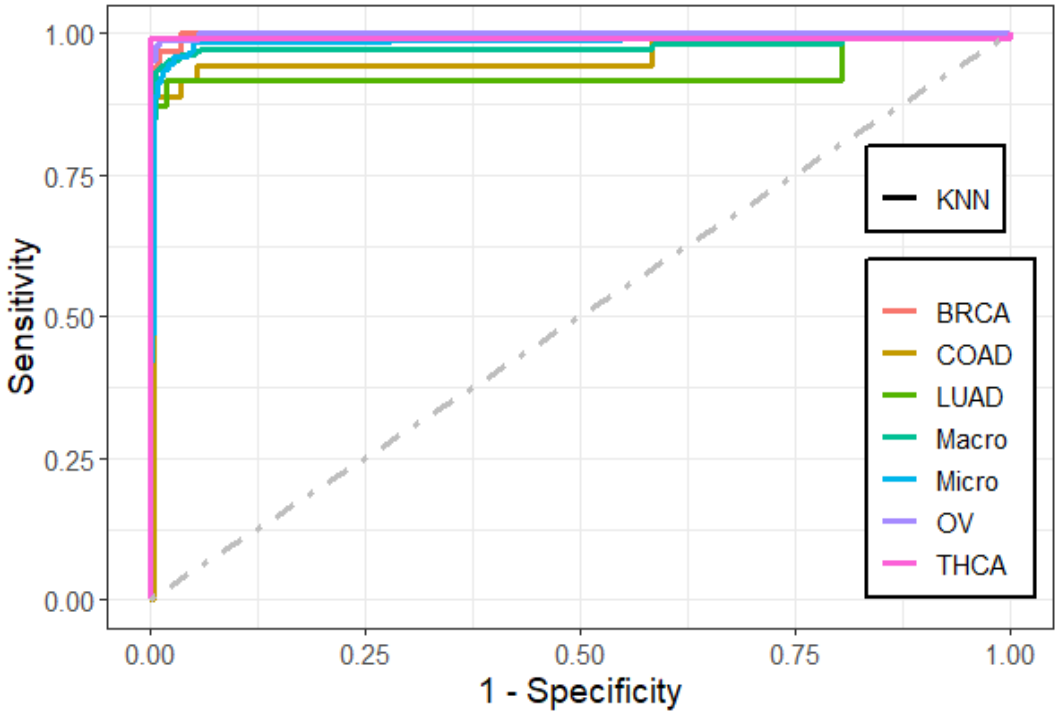


Figure 3.8: Multi-class ROC curves visualization for the KNN model based on under-sampling technique.

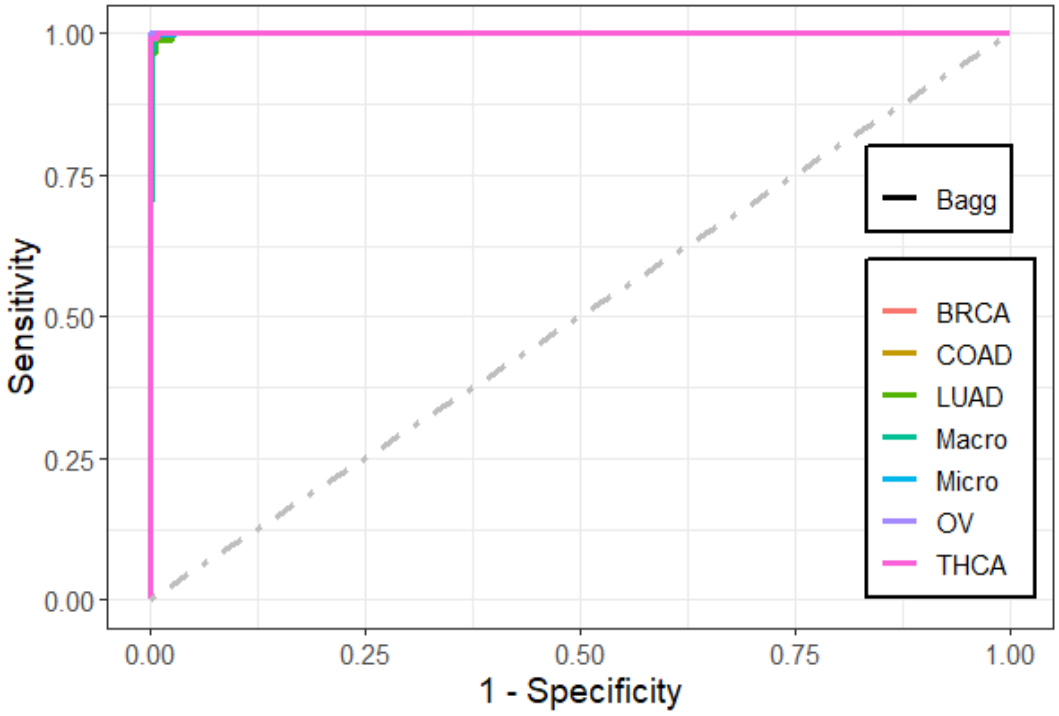


Figure 3.9: Multi-class ROC curves visualization for the bagging trees model based on under-sampling technique.

3.4.3 Predictive performance of the one-dimensional convolutional neural network model

The results that are presented in Table 3.4 show that the 1D-CNN model has a high performance when applied on the genes that are selected using LASSO (173 genes) where it achieved an average classification accuracy of 99.22%. These results also showed that the 1D-CNN outperformed the results of the machine learning methods that are presented in Table 3.2. It can be noted from the overlapped confusion matrix of the multiclass classification that the deep learning model classified the five categories of the cancers types using the 173 genes better than classifying these categories using the full list of genes $(14,899)$. The resulting precision, recall, and F1-score values are 99.32%, 99.09%, and 99.19%, respectively.

Table 3.4: The performance of the 1D-CNN model using early stopping regularization.

All (14,899 Genes)											
Performance Measures	**Folds**										**Overall**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
Accuracy	99.54	98.16	95.85	97.24	97.24	97.24	99.54	96.30	99.54	100	98.06
Precision	99.47	96.07	93.50	96.72	96.92	95.11	99.82	94.16	99.38	100	97.12
Recall	99.26	98.20	96.56	95.22	96.82	96.06	99.26	94.94	99.81	100	97.61
F1-Score	99.36	97.03	94.87	95.94	96.78	95.48	99.53	94.54	99.59	100	97.31
Reduced (173 Genes)											
Accuracy	98.62	99.54	99.08	98.62	99.54	100	99.07	99.54	98.61	99.54	99.22
Precision	99.46	99.31	99.10	98.99	99.82	100	98.48	99.29	98.92	99.82	99.32
Recall	97.97	99.82	99.10	98.39	99.29	100	98.72	99.81	98.52	99.26	99.09
F1-Score	98.68	99.56	99.10	98.65	99.54	100	98.57	99.54	98.69	99.53	99.19

Figures 3.10, 3.11, 3.12, and 3.13 show F1-measure and accuracy for training and validation when training our model using the full list of genes and the reduced genes with the early stopping approach. These figures indicate that the model can generalize very well since they become stable when the F1-measure and the accuracy are more than 99%. Figures 3.14 and 3.15 show the losses when using the full list of genes and the LASSO selected genes, respectively.

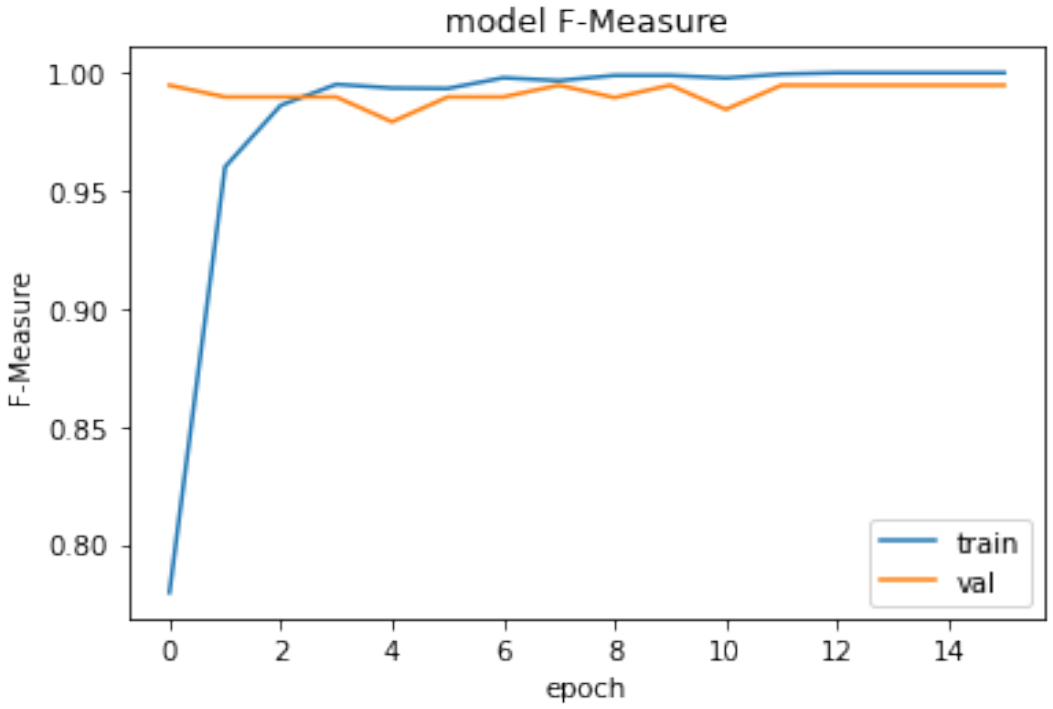


Figure 3.10: Training and validation F1 measure for the full list of genes with early stopping.

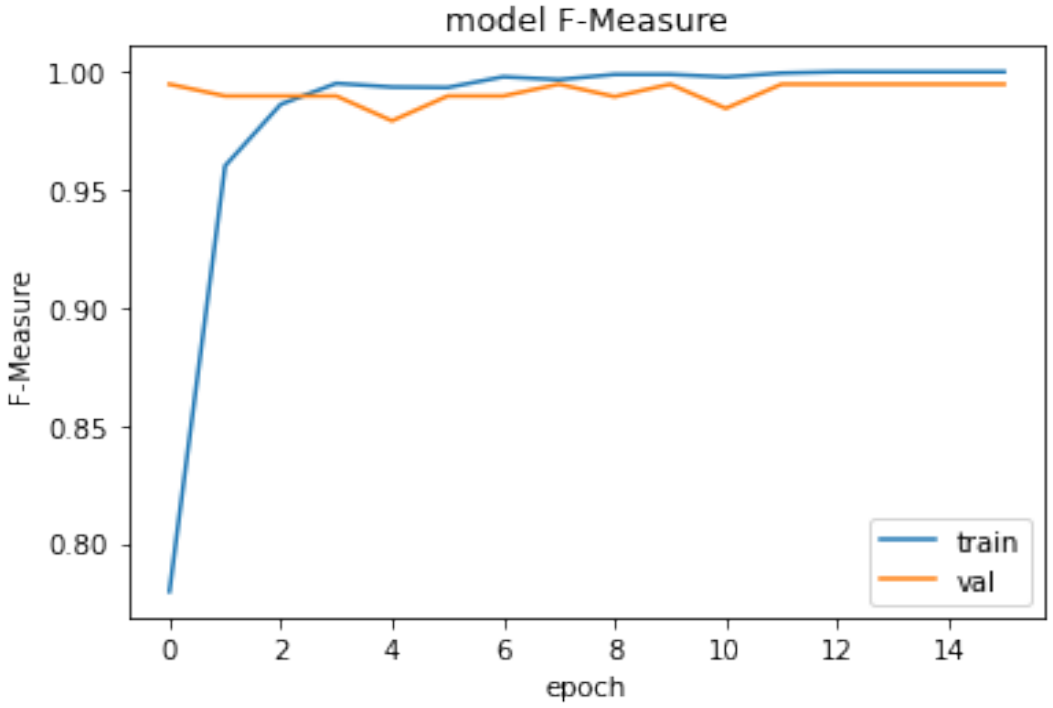


Figure 3.11: Training and validation accuracy for the full list of genes with early stopping.

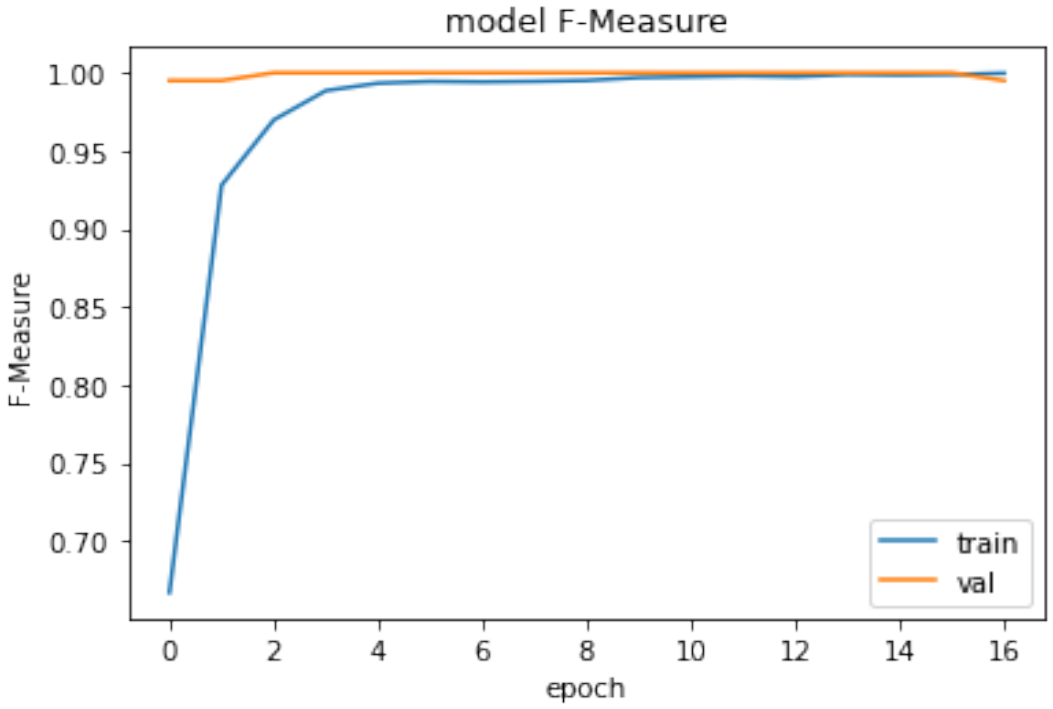


Figure 3.12: Training and validation F1 measure for reduced genes with early stopping.

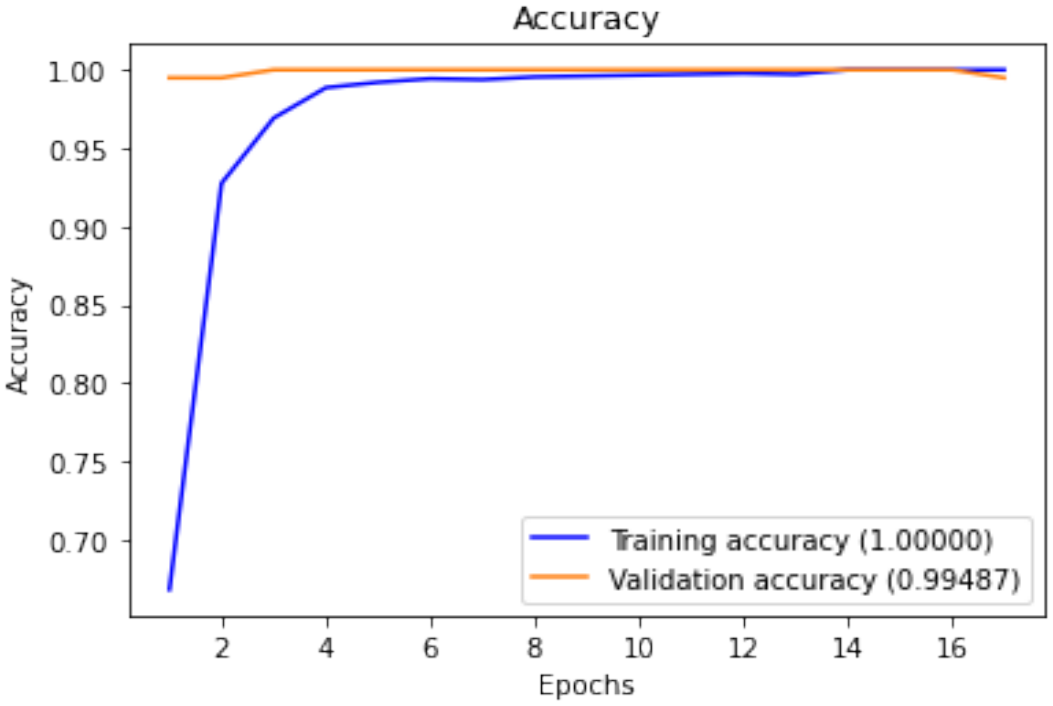


Figure 3.13: Training and validation accuracy for reduced genes with early stopping.

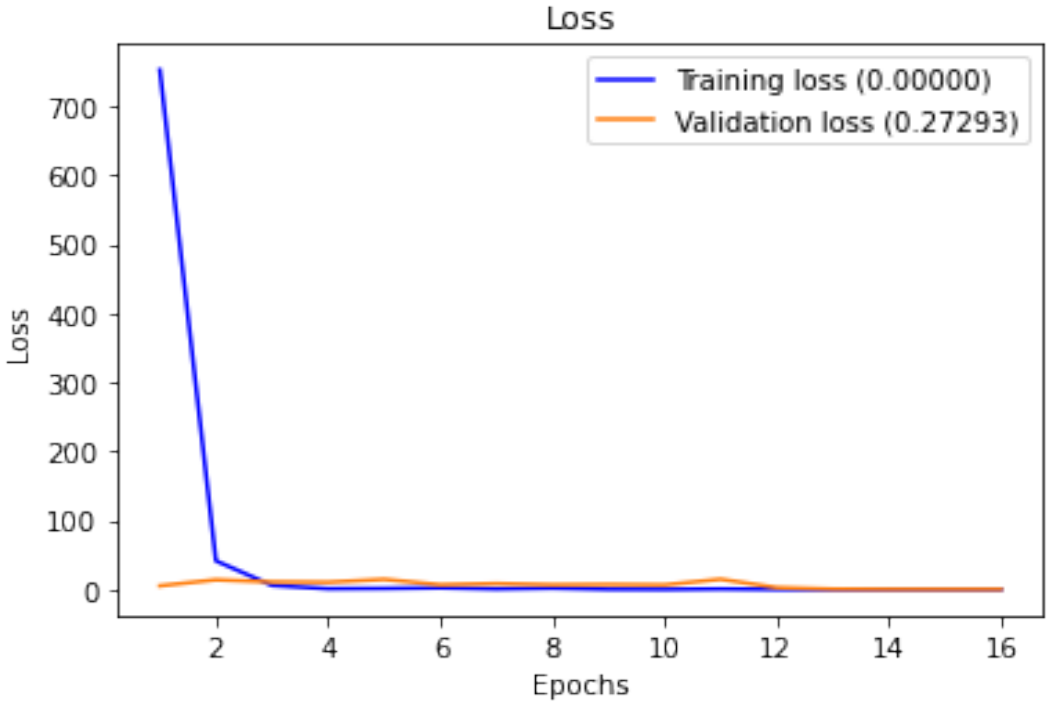


Figure 3.14: Training and validation loss for the full list of genes with early stopping.

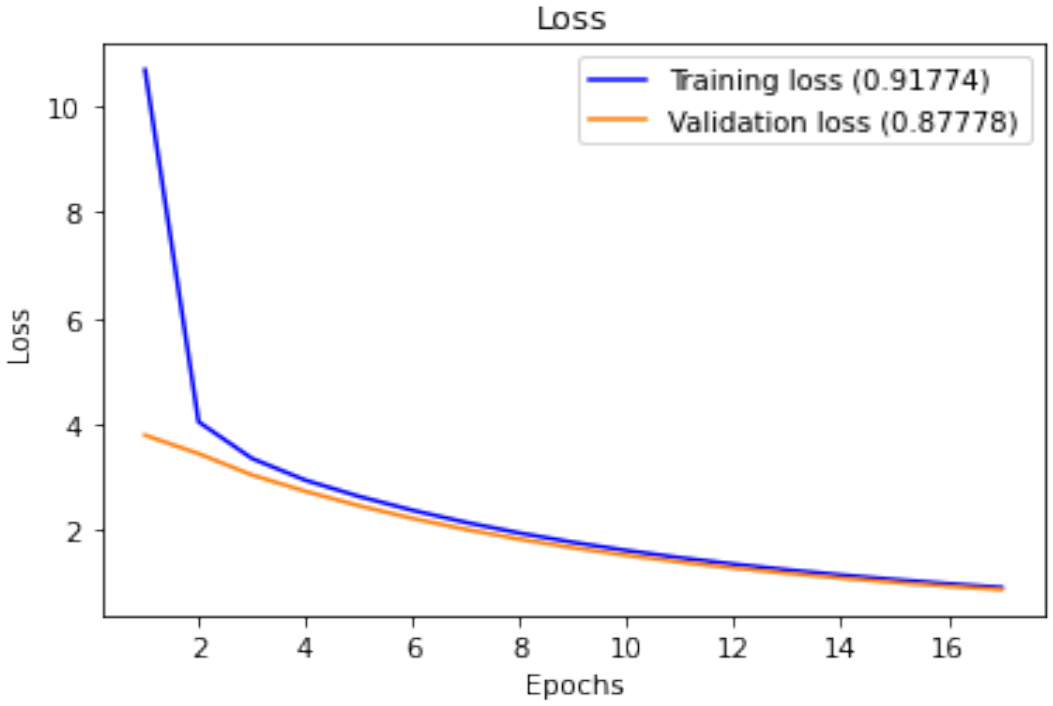


Figure 3.15: Training and validation loss for reduced genes with early stopping.

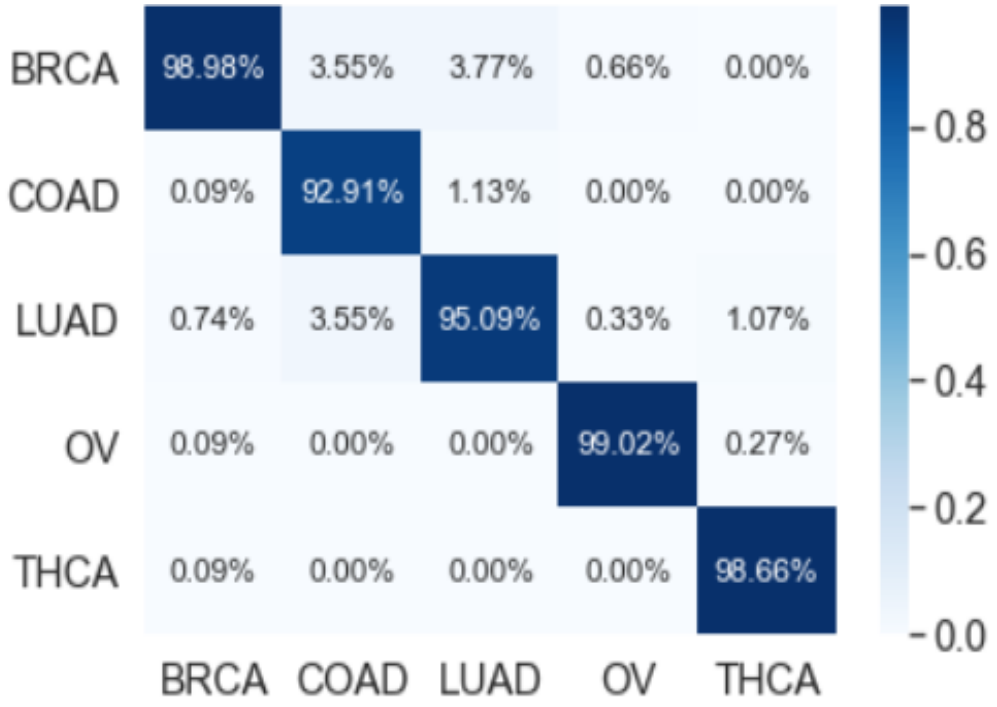


Figure 3.16: 10-folds overlapped confusion matrix (CM) for all $14,899$ genes.

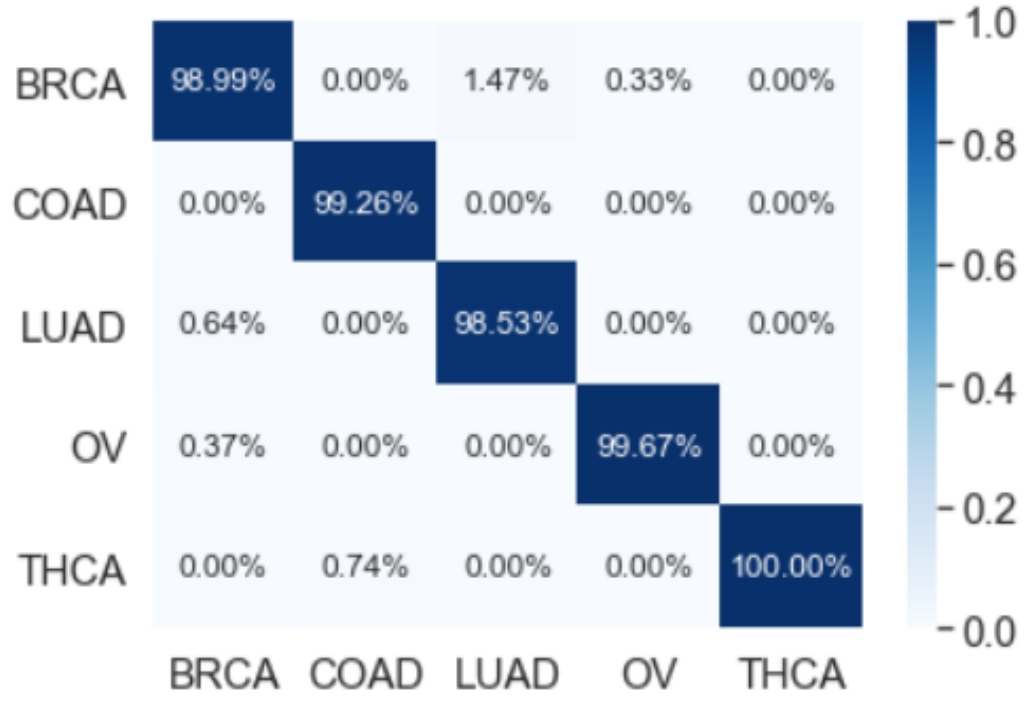


Figure 3.17: 10-folds overlapped confusion matrix (CM) for the reduced 173 genes.

The multi-class classification performance of the 1D-CNN model has been evaluated for each fold, and the average classification performance of the model is calculated. The overlapped confusion matrix (CM) is shown in Figs. 3.16 and 3.17 for all and reduced lists of genes, respectively. The overlapped CM is created using the sum of the ten separated confusion matrices. Thus, it is aimed to obtain an idea about the general perforations of the model.

Although we are using RNAseq data with a high number of genes, deep learning method outperformed the machine learning methods noting that a rigorous preprocessing step including a model-based approach using LASSO regression was applied to reduce the number of genes to be less than the number of observations.

The results that are presented in Table 3.5 below show that our proposed model has a high performance when applied on the genes that are selected using LASSO (173 genes) where it achieved an average precision, recall, and F1-Score of 99.55%, 99.29%, and 99.42% respectively. While the classification accuracy is 99.45% which is lower compared to accuracy of the full genes. These results also showed that our proposed model outperformed the results of the single 1D-CNN model and machine learning that are presented in Tables 3.2 and 3.4. In addition, Figs. 3.18 and 3.19 which is the overlapped confusion show that our proposed model has a better classification performance compared compared to the single 1D-CNN. Overall, our proposed model performance without using LASSO as a feature selection method is comparable to the performance with LASSO.

Table 3.5: The performance of the 1D-CNN model using early stopping regularization.

Performance Measures	Folds										Overall
	1	2	3	4	5	6	7	8	9	10	
All (14, 899 Genes)											
Accuracy	99.45	99.26	99.63	99.08	99.63	99.45	99.63	99.45	99.63	99.63	99.48
Precision	99.23	99.15	99.57	98.57	99.57	99.23	99.57	99.23	99.57	99.57	99.33
Recall	98.88	98.53	99.57	98.12	99.57	99.50	99.57	98.88	99.57	99.57	99.18
F1-Score	99.05	98.83	99.57	98.31	99.57	99.36	99.57	99.05	99.57	99.57	99.25
Reduced (173 Genes)											
Accuracy	99.45	99.26	99.26	99.26	99.45	99.45	99.45	99.63	99.82	99.45	99.45
Precision	99.58	99.31	99.13	99.31	99.58	99.60	99.58	99.65	99.93	99.79	99.55
Recall	99.19	99.12	99.31	99.12	99.19	99.38	99.19	99.47	99.72	99.19	99.29
F1-Score	99.38	99.22	99.22	99.22	99.38	99.49	99.38	99.56	99.82	99.49	99.42

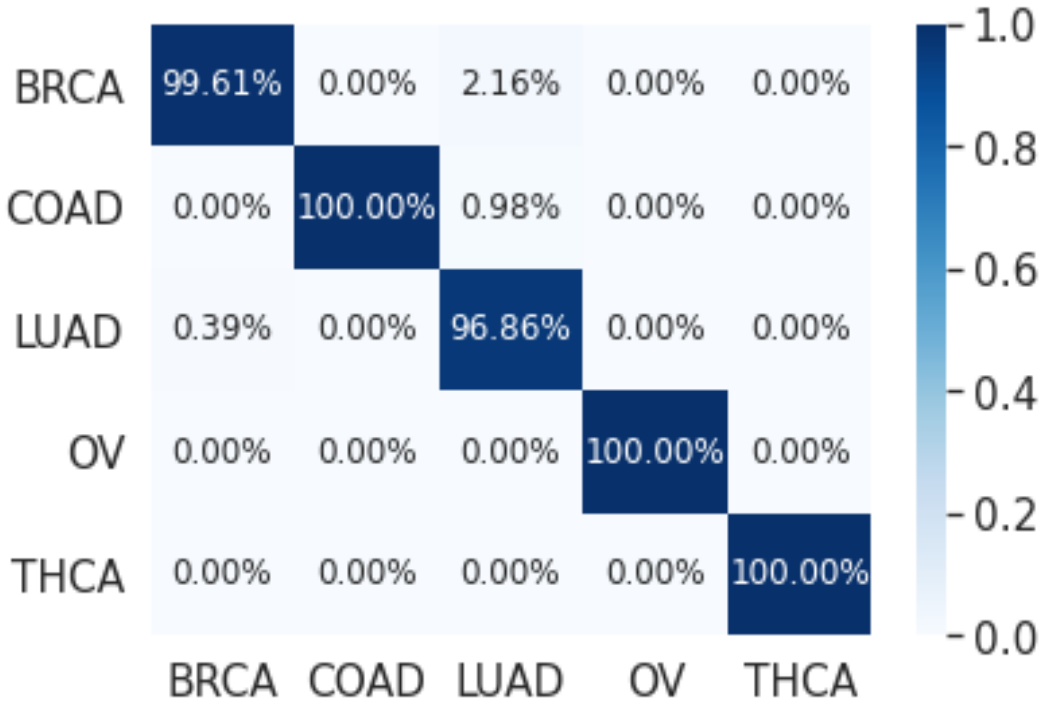


Figure 3.18: 10-folds stacking ensemble deep learning model overlapped confusion matrix (CM) for all 14, 899 genes.

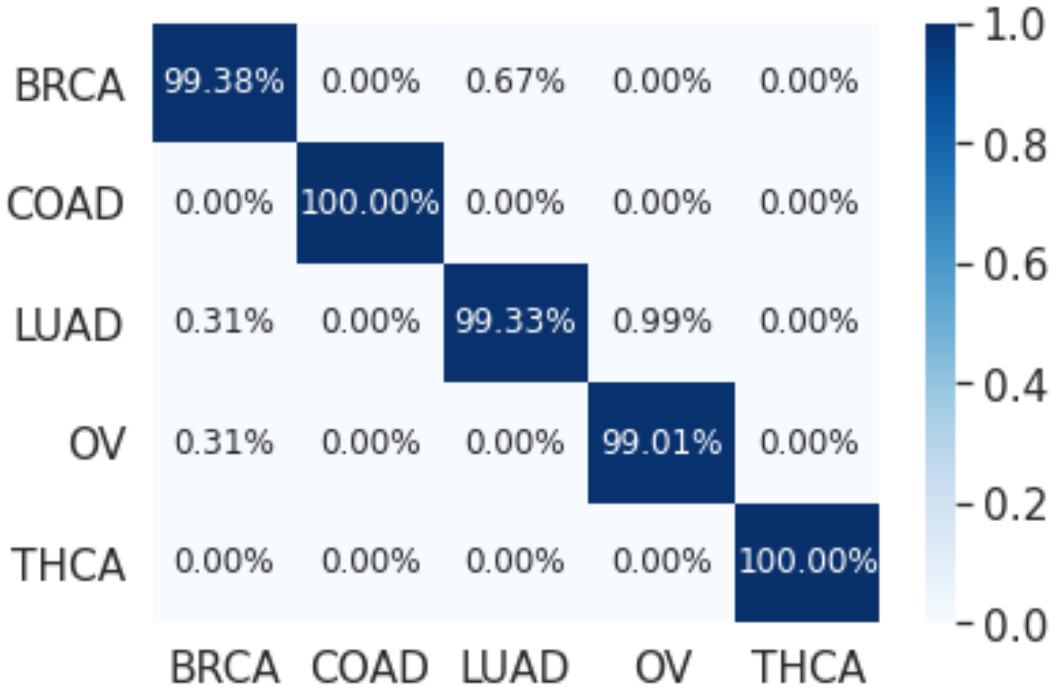


Figure 3.19: 10-folds stacking ensemble deep learning model overlapped confusion matrix (CM) for the reduced 173 genes.

A comparison of the methods was statistically conducted using the pairwise analysis test which produced pairwise statistical significance table of scores where the lower diagonal of the table shows p-values for the null hypothesis (distributions are the same), smaller p-value is indicative of a better model. The upper diagonal of the table presents the estimated differences in mean accuracy and kappa coefficient between the distributions. From Table 3.6 (under-sampling technique) we can see clearly of the fifteen pairwise comparisons of the six machine learning methods, there are nine comparisons showing statistically significant differences in terms of accuracy at the 0.05 level of significance. These differences are SVMR differed statistically to SVMP $p = 0.003$, ANN $p \leq 0.001$, and KNN $p \leq 0.001$. While SVML differed statistically to ANN $p = 0.009$, and SVMP differed statistically to ANN $p \leq 0.001$ and KNN $p \leq 0.001$. Moreover, ANN differed statistically to bagging trees $p \leq 0.001$, as well as KNN differed statistically to bagging trees $p = 0.004$.

Table 3.6: Pairwise statistical analysis test p-values and the estimated differences for the machine learning models (under-sampling technique).

Accuracy						
	SVMR	**SVML**	**SVMP**	**ANN**	**KNN**	**Bagging trees**
SVMR		0.015	-0.015	0.138	0.038	-0.003
SVML	1.00		-0.030	0.123	0.022	-0.019
SVMP	0.003	0.347		0.153	0.052	0.011
ANN	<0.001	0.009	<0.001		-0.101	-0.142
KNN	<0.001	1.00	<0.001	0.008		-0.041
Bagging	1.00	1.00	0.250	<0.001	0.004	
Kappa						
	SVMR	**SVML**	**SVMP**	**ANN**	**KNN**	**Bagging trees**
SVMR		0.024	-0.021	0.194	0.054	-0.005
SVML	1.00		-0.045	0.170	0.030	-0.029
SVMP	0.003	0.386		0.215	0.075	0.016
ANN	<0.001	0.010	<0.001		-0.140	-0.199
KNN	<0.001	1.00	<0.001	0.006		-0.059
Bagging	1.00	1.00	0.250	<0.001	0.004	

3.5 Discussion

We applied a novel stacking ensemble deep learning model to classify five common cancers among women: breast, colon adenocarcinoma, lung adenocarcinoma, ovarian, and thyroid cancers. The performance of the current proposed model is compared with the single 1D-CNN and machine learning methods that are mostly used in cancer types classification. We showed that the best machine learning average results were obtained using 173 genes based on the under-sampling

technique, while our proposed model has the highest performance based on the early stopping regularization. The improvement in accuracy was achieved by optimizing several parameters. We used LASSO as a feature selection technique with our proposed model to explore the integration of features selection method with a deep learning approach because features selection in deep learning is still unexplored area due to the black box nature of the deep learning methods. The results of the proposed model without using LASSO as a feature selection technique is comparable to the results with LASSO. This indicates that the 1D-CNN performs features selection through its layers. Bagging trees obtained excellent results, with a maximum accuracy of 99.2% among the machine learning models based on the under-sampling technique. In contrast, ANN showed the least accuracy of 80.7% for classifying the most common cancers among females. The SVM-P method showed performances that was close to the bagging trees method with an accuracy of 98.9% when we used the under-sampling technique. Overall, our results showed that SVM-R, SVM-L, SVM-P, ANN, KNN, and bagging trees were improved in performance if under-sampling is applied compared to over-sampling. We conclude that our proposed model is the best methods for the test dataset in this study. However, bagging trees is the best model among the machine learning models.

Overall, our proposed model surpassed the single 1D-CNN and the machine learning methods in the classification of common cancers among women. These findings are different from those reported in other studies (García-Díaz et al., 2020; Castillo et al., 2019; Ramaswamy et al., 2001). These differences can be explained by variations in the type of cancers studied and the methods used for feature/ gene selection. A study by Yang and Naiman (Yang & Naiman, 2014) introduced and validated a gene selection approach using machine learning methods but did not assess the performance of the machines. Our findings demonstrated that, our proposed model can achieve a higher performance on cancer tumor classification

using gene expression data. Both deep and machine learning methods and a combination of both can assist in predicting or detecting cancer susceptibility in the early stages and therefore, aid in designing early treatment strategies, and in turn increase survival of the high-risk women.

Because of the large number of genes in the gene expression data, we used LASSO regression as a rigorous feature selection method that reduced the dimensionality of the data sets (Fonti & Belitser, 2017; Ogutu et al., 2012). This process enabled us to retain the most important features (genes) for classification and prediction. In order to avoid over-fitting and the bias in the skewed class distribution we used over and under-sampling imbalance handling techniques, which improve the machine learning performance. In general, our results show that under-sampling technique improved the methods performance, and this is confirmed in previous studies (Galar et al., 2011; Rok & Lara, 2013; Van Hulse et al., 2007).

There were statistically significant differences ($p < 0.05$) between the machine learning methods, which demonstrates that the performance of the machines on cancer classification is not the same. However, deep learning methods outperformed the machine learning methods in cancer classification, which is similar to a previous study (Tabares-Soto et al., 2020). Overall, the accuracy of our proposed model on the full features and on the features that are selected using LASSO are 99.48% and 99.45%, respectively, which are 5.05% and 5.02% higher than accuracy obtained by (Tabares-Soto et al., 2020) which is 94.43%. We note that Tabares-Soto et al. (Tabares-Soto et al., 2020) used microarray gene expression data, focusing on 11 type of cancers for both males and females, compared to RNASeq data used in this study to classify five common cancers among females. This study also did not consider class imbalance handling methods as applied in the current study and had 12-times lower sample size ($n = 174$) than in our study ($n = 2166$). With larger sample size, more samples are available to train the models. These

issues were, therefore, likely to affect the reliability of findings and potentially affecting the performance of the methods. Our study was limited to the gene expression profiles from RNASeq data. However, Lee and co-workers (Lee et al., 2019) used several features such as mutation profiles and mutations rates. They evaluated different machine learning and feature selection methods using RNASeq data from 31 cancer types. The highest accuracy they obtained was 84%. Thereafter, they reduced the number of cancers to the six most common types and obtained an accuracy of 94%, which is low compared to our proposed deep learning model.

Our proposed model has a very high achievement in classifying the five common cancers among women and that may potentially improve the multi-class identification (Ramaswamy et al., 2001). In addition, this study is first of its kind to classify cancer tumors using RNAseq data. However, multi-class cancer classification using gene expression is not a substitute to the traditional diagnosis (Ramaswamy et al., 2001), but advances in classification algorithms or methods may provide a more accurate and biologically meaningful classifications and inform future studies. Moreover, a more pressing classification problem may be that of discriminating between cancer sub-types within the same type than between cancer types. However, we postulate that the methods covered in this paper are directly applicable to this problem.

3.6 Conclusion

In this work, we proposed a stacking ensemble deep learning model as a multi-class classifier to classify five most common cancers among women, that is, breast, colon adenocarcinoma, lung adenocarcinoma, ovarian, and thyroid cancer, using RNASeq gene expression datasets for each cancer tumor. Tumor classification using RNASeq data is more accurate and available compared to microarray data. We used LASSO as a feature selection method and compared the performance of our proposed method with a stand alone deep learning and machine learning

methods. We conclude that our proposed model achieved the highest performance compared to the single 1D-CNN and the machine learning methods. Our proposed model is, therefore, capable of correctly classifying all the observed positive cancer cases. The proposed model can help improve the detection and diagnosis of cancer susceptibility among women in the early stages, inform decision on early intervention, and hence improve survival. Future research should consider the potential effects of using many feature types such as methylations, mutations, proteins, Single nucleotide polymorphisms (SNPs), etc, to be integrated with RNASeq data. Future work will also consider improvements on the stacking ensemble problem including statistical properties to improve inference.

Chapter 4

Paper III: Predictors of Colorectal Cancer Survival using Cox Regression and Random Survival Forests Models Based on Gene Expression Data

This chapter addresses the following objective:

- Determine the predictors and genes of CRC using Cox PH and RSF models, including handling missing values in the dataset.

4.1 Abstract

Understanding and identifying the markers and clinical information that are associated with colorectal cancer (CRC) patient survival is needed for early detection and diagnosis. In this work, we aimed to build a simple model using Cox proportional hazards (PH) and random survival forest (RSF) and find a robust signature for predicting CRC overall survival. We used stepwise regression to

develop Cox PH model to analyse 54 common differentially expressed genes from three mutations. RSF is applied using log-rank and log-rank-score based on 5000 survival trees, and therefore, variables important obtained to find the genes that are most influential for CRC survival. We compared the predictive performance of the Cox PH model and RSF for early CRC detection and diagnosis. The results indicate that *SLC9A8*, *IER5*, *ARSJ*, *ANKRD27*, and *PIPOX* genes were significantly associated with the CRC overall survival. In addition, *age*, *sex*, and *stages* are also affecting the CRC overall survival. The RSF model using log-rank is better than log-rank-score, while log-rank-score needed more trees to stabilize. Overall, the imputation of missing values enhanced the model's predictive performance. In addition, Cox PH predictive performance was better than RSF.

4.2 Introduction

Colorectal cancer (CRC) is the second leading cause of mortality in women and third in men (Favoriti et al., 2016). The American cancer society estimate, about 1 in 23 men and 1 in 25 women develop colorectal cancer in their lifetime (Society, 2020). Globally, there were about 19.3 million new cancer cases in 2020 alone, while close to 10 million deaths were recorded due to cancer (Sung et al., 2021). CRC represents 9.4% of cancer deaths and 10% of newly diagnosed cancer cases (Sung et al., 2021). The incidence and mortality in males are 10.6% and 9.3%, respectively, while the incidence and mortality in females are 9.4% and 9.5%, respectively (Sung et al., 2021). Early detection of CRC can reduce mortality due improved chemotherapy regimens and surgical techniques (Dai et al., 2020; Bray et al., 2018; Stintzing, 2014). The prognosis and survival of early intervention with CRC patients are linked with tumor staging, where early diagnosis of the tumor is more likely to be curable (Bian et al., 2019). The 5-year relative survival rates for patients with localized CRC was 91% in the USA between 2010 and 2016 (ACS, 2021). However, the 5-year relative survival rates of CRC cases at regional and distant stages are 72% and 14%, respectively (ACS, 2021). The main characteristics of the

CRC are that it has high inter-patient and intra-tumor heterogeneity. Other factors such as environment, lifestyle, and diet can lead to further heterogeneity in the CRC occurrence and progression (Molinari et al., 2018; Bramsen et al., 2017; Ogino et al., 2018). This heterogeneity leads to variations in response to treatment between individuals. Determining the molecular markers is clinically essential to help detect and precisely predict the prognosis of patients with CRC.

Researchers have developed many methods to determine the prognostic molecular markers to early detect and predict the prognosis of patients with CRC. These methods include univariate and multivariate Cox proportional hazard models, elastic net estimation, and random forests for survival prediction (Dai et al., 2020; Bian et al., 2019; Aziz et al., 2016; Pan et al., 2019; Martinez-Romero et al., 2018; Yan et al., 2012). Previous studies such as, Abdul Aziz et al. (Aziz et al., 2016) analyzed the CRC death using the Cox proportional hazard model, and they reported a 19 gene signature that could predict the survival of CRC patients with Dukes' B and C stages. In their work, Abdul Aziz et al. used SAM, *limma*, and t-test to identify the most significant genes based on microarray gene expression data. Dai et al. (Dai et al., 2020) conducted a survival analysis using univariate and multivariate Cox models based on three microarray datasets from GEO and one dataset from the TCGA database. They used the DEGs from each of the three microarray datasets, and they identified 105 mutual DEGs based on the intersection of the three DEGs lists. They conducted a protein-protein interaction network (PPI) of the DEGs, and they identified hub genes. To investigate the 44 hub genes' prognostic values in CRC, they conducted a survival analysis using the sample splitting and Cox regression models based on the TCGA dataset. Their results showed that two down-regulated and two up-regulated hub genes were significantly associated with the CRC patients' overall survival. Bian et al. (Bian et al., 2019) analyzed data from four microarray datasets and identified DEGs from each of them. They identified the common genes across the four datasets, and this way, they obtained 53 genes.

Then they utilized PPI, which identified ten essential genes according to their degree value, betweenness centrality, and closeness centrality. They used gene expression profiling interactive analysis (GEPIA) to apply survival analysis using the log-rank test based on the expression levels. Their results showed that four low expressed genes out of the ten genes were significantly related to unfavorable prognosis in the patients with CRC. Martinez-Romero et al. (Martinez-Romero et al., 2018) identified a new set of gene markers associated with CRC to predict tumor progression and evolution towards inferior survival stages based on an integrated gene expression dataset of 1273 CRC samples. They compared the early and late stages of CRC using *limma* to identify the genes (2707 DEGs) that had a significant effect on CRC tumor progression. Then, they applied Kaplan-Meier to rank the genes based on the non-parametric log-rank test. Their results identified 429 essential genes in which overexpression is related to low survival rate and 336 crucial genes in which repression is associated with inferior survival. They validated the top 5 genes using an external cohort study and presented a good separation of the CRC samples into two low and high-risk groups.
A study by Pan et al. (Pan et al., 2019) proposed a predictive model based on RNASeq gene expression data. Their model uses the differentially expressed genes (DEGs) profiles. These profiles were obtained using the univariates and multivariate Cox regression, which was used to compare TNM stages to assess their predictive survival accuracy. Their results showed that 10 DEGs had a significant effect on CRC survival. Yan et al. (Yan et al., 2012) implemented random forests to identify biomarkers associated with survival in CRC based on a set of oligonucleotide microarray data. Their results showed that four genes had the potential to predict CRC survival.

To the best of our knowledge, RSF has not been used with gene expression data in the previous studies to predict CRC survival. The gene expression data is characterized by the problem of the curse of dimensionality and collinearity. To

overcome this problem, the CRC survival is predicted based on selecting the differentially expressed genes (DEGs) in colorectal cancer that was based on the three-mutation status (KRAS, BRAF, and TP53) where they serve as a predictive biomarker of response to treatment in CRC. We assume that complex interaction between multiple DEGs contributes to prognostic survival differences between wild-type and mutant patients with CRC.

We developed and compared Cox proportional hazard (Cox PH) model and random survival forests (RSF) in predicting CRC survival and associated biomarkers using a public genome database from Gene Expression Omnibus (GEO). The aim was to assess the CRC survival predictors accounting for missing data based on the gene expression data. We selected 54 common differentially expressed genes from three mutations (KRAS, BRAF, and TP53), using the complete case samples, and performed analysis using Cox PH and RSF models before and after imputation.

4.3 Materials and Methods

4.3.1 Dataset

The dataset with accession number GSE39582 (Marisa et al., 2013), was downloaded from Gene Expression Omnibus (GEO) public database (https://www.ncbi.nlm.nih.gov/geo/) using the BRB-ArrayTools software (https://brb.nci.nih.gov/BRB-ArrayTools/). This dataset has 54675 probes taken from 566 samples with colon cancer and 19 non-tumor samples. Usually, the gene expression data includes noisy and or irrelevant genes. Therefore, performing data cleaning and feature (genes) selection are essential steps that should be applied before modeling the data. A pre-processing step was applied to prepare the dataset for modeling. These pre-processing steps are log2 transformation, quantile normalization, gene filtration, and differentially expressed genes analysis using a two samples t-test. Filtration is a process in data cleaning used to eliminate

insufficiently expressed probes and those with excessive missing expression levels across the samples (Simon et al., 2007; Chaba, 2016; Bolstad et al., 2003; Mohammed et al., 2018). On the other hand, quantile normalization and log2-transformed steps to eliminate the variation between samples. BRB-ArrayTools is used to implement the filtration and normalization of the dataset. The two-sample t-test, with the 0.001 significance level threshold, was used for gene selection to provide informative genes for building survival models. The overall procedures that we followed in our analysis are summarized in Figure 4.1.

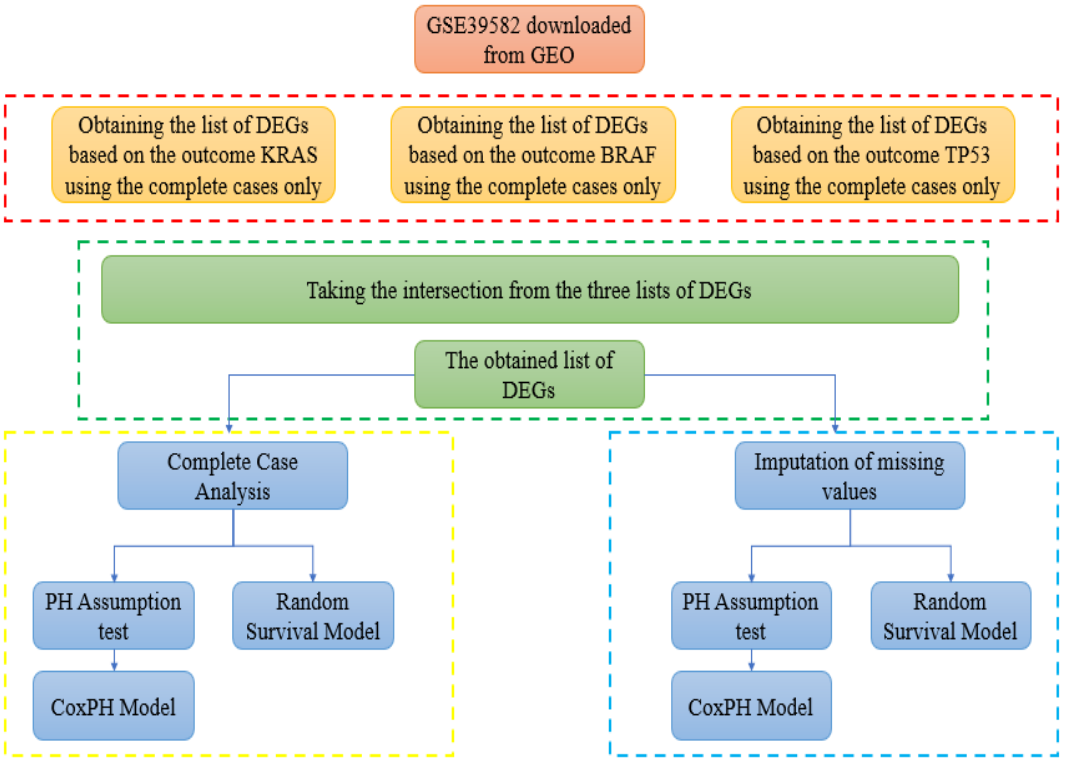


Figure 4.1: Flow-chart of the procedure followed in the pre-processing and analysis of the dataset.

4.3.2 Statistical Analysis

We analyzed the gene expression data using the *R* version (R-4.0.4). Summary statistics of the gene expressions are depicted in the supplementary file (see Appendix 5.4). These statistics include the minimum, maximum, means, and standard deviations of the expression levels. We used frequency and percentages

for the categorical data representing the clinical information, as shown in 4.1. The statistical analysis was conducted in three phases; the first phase is the complete case analysis, followed by imputation of missing values in the outcome based on the covariates and an appropriate imputation model. Then we applied survival analysis on the complete case and imputed datasets. The survival analysis results on these two datasets were compared to evaluate the precision of estimates. Two separate models were fitted before and after imputations; the first is the Cox regression model, while the second is the random survival forests with log-rank and log-rank-score split rules. The missing values were assumed to be missing at random (MAR), where the probability of data being missing does not depend on the unobserved data, conditional on the observed data (Pedersen et al., 2017; Rezvan et al., 2015; Mboya et al., 2020, 2021); consequently, the genes and other covariates in the dataset were used to predict missingness.

4.3.2.1 Complete case analysis

The filtration step resulted in 18865 out of 54675 probes. These 18865 probes were used for further reduction analysis using a t-test. To find the differentially expressed genes (DEGs) that discriminate between the mutant and wild-type mutation, we used the three mutation types, *KRAS*, *BRAF*, and *TP53*. We created three different datasets using the 18865 probes with each of the three mutation types based on these three mutation types. First, we removed the samples with missing values for each of the three datasets according to their clinical outcome. Then, we calculated the correlation matrix for the gene expression data and filtered out one gene from every two genes that show a correlation coefficient greater than 0.6. Subsequently, we extracted three DEGs lists from all three datasets using a two-sample t-test based on 0.001 thresholds. Ultimately, from the three lists of DEGs, there were 54 common genes (see Appendix 5.4). Also, we used the common samples across the three datasets to produce the complete cases in one dataset. The samples with missing or zero values in the event status and time variables were

removed. We then converted the five TNM stages into a new categorical variable with two stages (Early and Late), where stages four and five were combined to give the late category. Finally, we used the obtained data for finding the most significant gene markers that may predict survival for CRC patients. Table 4.2 provides a concise summary of the pre-processed data.

Table 4.1: Clinical characteristics of colorectal cancer patients (N=307).

Variable	Frequency (n)	Percentage (%)
Age at diagnosis in years: Mean (SD*)	66.8 (13.2)	
KRAS Mutation		
Mutant	123	40
WildType	184	60
BRAF Mutation		
Mutant	25	8
WildType	282	92
TP53 Mutation		
Mutant	166	54
WildType	141	46
Tumor Location		
Proximal	124	40
Distal	183	60
Cancer stage		
Early	156	51
Late	151	49
Sex		
Female	137	45
Male	170	55
Molecular subtype		
C1	65	21
C2	49	16
C3	43	14
C4	29	9
C5	29	9
C6	36	12

*SD: Standard deviation.

Table 4.2: Summary of the filtered datasets and the pre-processing steps.

Dataset ($GSE39582$)*		Number of samples	Complete cases	Common samples	Total number of genes	After filtration	Uncorrelated genes	DEGs (t-test)	Common genes
	KRAS		545					711	
Clinical outcomes	BRAF	585	512	307	54675	18865	13827	2388	54
	TP53		351					629	

* Three datasets with the same covariates and different clinical outcome.

4.3.2.2 Multiple imputations of the missing values

To compensate for the missing data, we used the *R* package "*mice* (Multivariate Imputation by Chained Equations)", which impute the missing values in the covariates. The *mice* package takes care of uncertainty related to missing values (Mboya et al., 2020, 2021; Jakobsen et al., 2017). It assumes that the missing values are missing at random (MAR) see Figure 4.2, where the probability of missing data does not depend on the unobserved data, conditional on the observed data (Pedersen et al., 2017; Rezvan et al., 2015; Mboya et al., 2020, 2021). The *mice* package uses the genes and other covariates in the dataset to predict missingness. The missingness pattern in the data is assumed to be non-monotone. In this pattern, some subject values can be observed again after missing values happen (Mboya et al., 2020, 2021; Jakobsen et al., 2017). For this missing data pattern, it is recommended to use the chained equations (fully conditional specification (FCS)) (Azur et al., 2011), or the Markov Chain Monte Carlo (MCMC) method to impute missing values (Jakobsen et al., 2017).

We used FCS to handle the missing values in our dataset implemented in the *mice* package in *R* using a random forest model. The FCS is considered a powerful and statistically valid method for creating imputations in both categorical and continuous variables (Azur et al., 2011). We generated 5 imputed datasets using random forest (rf) imputations after 100 iterations (imputation cycles). We used 1051991 as a random seed to replicate imputation results each time a multiple imputation analysis was performed. In addition, we followed the procedures

indicated by the work of Sterne et al. (Sterne et al., 2009) for reporting and analysis of missing data. *KRAS*, *BRAF*, *TP53*, and the event status were imputed as binary, while time and age imputed as numeric variables. The rest of the variables did not contain any missing values, and were used as auxiliary variables in the imputation model. Overall, firstly we performed a complete case analysis using Cox PH and random survival forests models. Thereafter, we compared the final models from this analysis to those from the multiply imputed dataset.

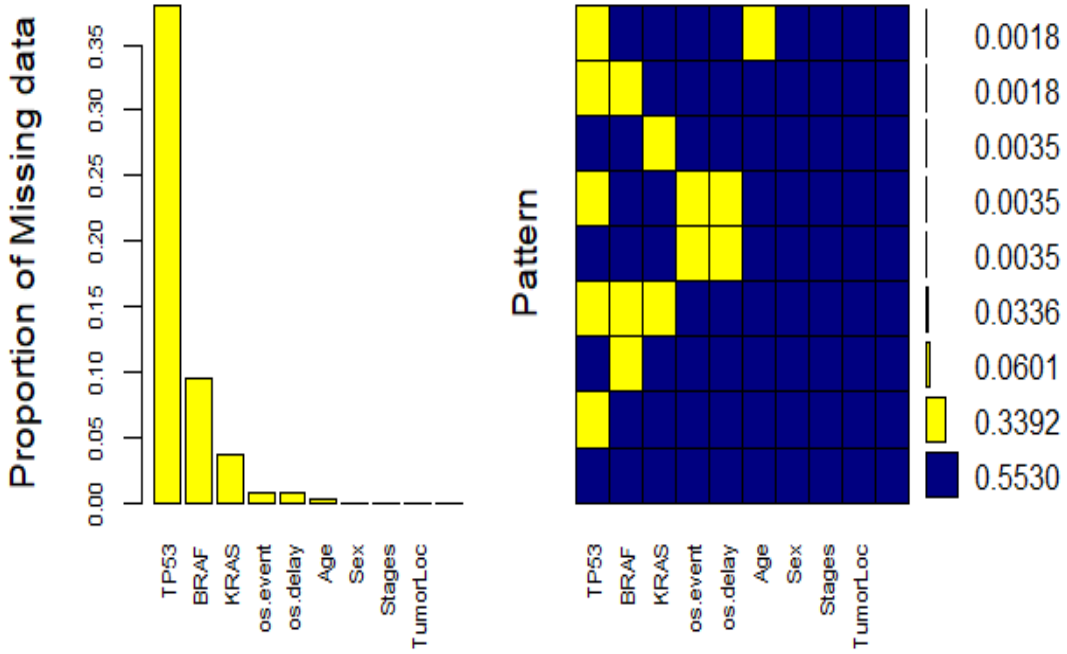


Figure 4.2: Proportion and patterns of missing values in the clinical characteristics available in the GSE39582 dataset.

4.3.3 Experimental setup

To evaluate the different methods, the resulting dataset was divided into training set (80%) and testing set (20%). The training set was then divided into 10 subsets to train the methods using 10-fold cross validation approach to avoid overfitting. In the 10-fold cross-validation approach the integrated brier scores (IBS) is calculated on each fold left-out while the model is trained on the other 9 folds. Finally, the trained model

is tested on the testing set. The model performance was measured using prediction error curve (pec).

4.3.4 Statistical methods

4.3.4.1 Cox Proportional Hazard Model (Cox PH)

Cox proportional hazard model is the most widely used statistical model for modeling time to event data (Bradburn et al., 2003). The Cox PH evaluates the association of the survival time of patients and one or more predictors/genes variables. The Cox PH model relates the effect of predictors which include genes in our case to the rate or hazard of occurrence of an event such time to infection, death, recurrence of a condition at a certain point of time , this rate is generally referred as the hazard rate (Ajagbe et al., 2014; Kleinbaum & Klein, 2010). In order to estimate the association of the gene expression levels and the survival time, consider n cancer samples say from sample $i = 1, 2, \ldots, n$ and $g_i = (g_{i1}, g_{i2}, g_{i3}, \ldots, g_{ip})$ is a vector of p genes expression level. The i^{th} patient survival data can be represented by $(T_i, \delta_i, g_{i1}, g_{i2}, g_{i3}, \ldots, g_{ip})$, where $i = 1, 2, \ldots, n$; T_i and δ_i indicate the survival time and the censor status respectively. The Cox PH model is mathematically represented as follow

$$h_i(t) = h_0(t)e^{\boldsymbol{\beta}' \boldsymbol{g}_i}, \tag{4.1}$$

where the elements of the vector $\boldsymbol{\beta}'$ represent the regression coefficients and the elements of the vector $\mathbf{g}_i$ are the covariates (genes). The baseline hazard function $h_0(t)$ is unspecified and non-parametric function of an individual with all expression levels equal to zero (Aziz et al., 2016; Myte, 2013). The model has a parametric part specified by the linear predictor and assumed to be proportional to the non-parametric baseline hazard. This means that for two individuals, i and j, the hazard ratio is

$$\frac{h_i(t)}{h_j(t)} = \frac{e^{\beta' g_i}}{e^{\beta' g_j}}. \tag{4.2}$$

The hazard ratio is assumed to be independent of time t. The maximum partial likelihood method used to estimate the Cox PH model parameters is given by

$$L(\beta) = \prod_{r \in E} \frac{e^{\beta^T g_r}}{\sum_{j \in R_r} e^{\beta^T g_j}}, \tag{4.3}$$

where E indicates the indices of the events (e.g., deaths) and R_r represents the vector of indices of the individuals at risk at time $t_r - 0$. The results of the Cox PH model are easy to interpret, however, there are key assumptions needed such as linearity and proportional hazards. We used *survival* and *survminer* packages to implement Cox PH model in *R*.

Moreover, we performed the stepwise regression for developing the Cox PH model at a 5% threshold level to find a simple model that shows the essential genes (markers) and clinical covariates correlated with the CRC. At each time, we remove the genes/ covariates that are not significant at $\alpha = 0.05$ level of significance. Thereafter, we tested for the PH assumption, and the integrative analysis of the CRC data showed five genes (markers) that passed the PH assumption test. Thereafter, we used the five genes and the other clinical information to fit the Cox PH model.

4.3.4.2 Random Survival Forests (RSF)

Random survival forests are an ensemble of trees and a non-parametric method constructed by bagging of classification trees for right censored data (Ishwaran et al., 2008; Wang & Li, 2017). The RSF are an extension of the random forests method proposed by Breiman et al. (Breiman, 2001a). The RSF works on high-dimensional data where the number of covariates exceeds the number of the observations, also it can handle data that consist of complex and non-linear

relationships between the dependant and the independent variables and when the covariates violate the proportional hazard assumption (Shu, 2019). There are several advantageous of using the RSF method, such as, it is not based on any model assumption compared to Cox PH model. It seeks to find a model that best represent the data in the case of limited survival data, in addition, it can handle high-dimensional data unlike Cox PH, and it is robust to outliers in the explanatory variables (Wang & Li, 2017). RSF employ two steps of randomizations which are to grow the tree a bootstrap sample is randomly drawn and for splitting at each node of the tree, a random subset of covariates is selected. The two steps of the randomization help in decorrelate the tree (Mohammed et al., 2018; Wang & Li, 2017). The RSF implemented using the *randomForestSRC* package in *R* (Ishwaran et al., 2021).

Random Survival Forests Algorithm We used the RSF algorithm that was introduced in the work of Ishwaran et al. (Ishwaran et al., 2008) as shown below:

1. For i in $1 : ntrees$

   (a) Draw bootstrap samples from the original total number of samples. For each bootstrap exclude approximately 37% of the samples as out-of-bag (OOB) samples.

   (b) Build a survival tree for every bootstrap samples by recursively repeating the following steps for each node in a tree.

      i. Randomly select v genes at random from the p genes ($v = \sqrt{p}$).

      ii. To split the node, pick the best gene among the v genes, that maximizes survival differences between daughter nodes. We used log-rank and log-rank-score splitting rules as measures of survival differences.

      iii. Produce the tree to full size under the constraint that a terminal node should have no less than $d_0 > 0$ unique deaths.

iv. Calculate a cumulative hazard function (CHF) for every tree. Average the CHF for all the $ntrees$ trees to find the ensemble CHF.

v. Calculate the OOB prediction error for the ensemble CHF, using OOB samples.

Once the survival tree is built, the ends of the tree are called the terminal nodes. Assume, the terminal node is h and $t_{n,h}$ is the individual's death time at node h, $d_{n,h}$ is the number of deaths, and $M_{n,h}$ is the number of individuals at risk at time $t_{n,h}$. Therefore, the cumulative hazard function (CHF) can be estimated using the Nelson-Aalen estimator (Nelson, 1972) as follows

$$\hat{H_h}(t) = \sum_{t_{n,h} \leq t} \frac{d_{n,h}}{M_{n,h}}. \tag{4.4}$$

The CHF will be calculated for all the terminal nodes. The CHF for new observation i given a vector of genes as a covariate $\mathbf{g}_i$, can be calculated for one tree as follows

$$\hat{H}_h(t|\mathbf{g}_i) = \hat{H}_h(t), \quad \text{for } \mathbf{g}_i \in h. \tag{4.5}$$

To compute an ensemble CHF, the average of the $ntrees$ trees is calculated, and the bootstrap ensemble CHF for an observation i is

$$\hat{H}_e(t|\mathbf{g}_i) = \frac{1}{ntrees} \sum_{b=1}^{ntrees} \hat{H}_b(t|\mathbf{g}_i). \tag{4.6}$$

Let,

$$I_{i,b} = \begin{cases} 1, & \text{if } i \text{ is an OBB observation for } ntrees \text{ training sample.} \\ 0, & \text{Otherwise.} \end{cases}, \tag{4.7}$$

then the OOB ensemble CHF for an observation i is given by

$$\hat{H}^*_e(t|\mathbf{g}_i) = \frac{\sum_{b=1}^{ntrees} I_{i,b} \hat{H}^*_b(t|\mathbf{g}_i)}{\sum_{b=1}^{ntrees} I_{i,b}}, \tag{4.8}$$

therefore, $\hat{H}_e^*(t|\mathbf{g}_i)$ is an average over the training samples where i is an OOB observation.

Log-rank split rule

The log-rank split-rule is a measure of a node separation which helps in determining the best split for that node (Ciampi et al., 1987). Let h be a node of a tree, and let there are be n individuals with this node. Suppose (T_1, σ_1), (T_2, σ_2), $\ldots$, (T_n, σ_n) are the survival outcomes corresponding to the n individuals. Thus, the best split at node h on covariate x at split point c, is the one that maximize the log-rank statistic between the two daughter nodes (Ishwaran et al., 2008) given as follow

$$L(x,c) = \frac{\sum_{i=1}^{N}\left(d_{i1} - Y_{i1}\frac{d_i}{Y_i}\right)}{\sqrt{\sum_{i=1}^{N}\frac{Y_{i1}}{Y_i}\left(1-\frac{Y_{i1}}{Y_i}\right)\left(\frac{Y_i-d_i}{Y_i-1}\right)d_i}}. \tag{4.9}$$

The aim is to maximize the log-rank statistic by finding values of x and c that maximize $L(x,c)$. Specifically, we are looking to find a predictor x^* and c^* such that $|L(x^*, c^*)| \geq |L(x, c)|$ for every x and c. This process is repeated at every node until the terminal node is reach.

Log-rank-score split rule

The log-rank-score split rule is a version of the log-rank-score split rule (Nasejje et al., 2017). Consider $r = (r_1, r_2, \ldots, r_n)$ as a vector that ranks the survival times $(T, \delta) = ((T_1, \delta_1), (T_2, \delta_2), \ldots, (T_n, \delta_n))$ (Nasejje et al., 2017; Hothorn & Lausen, 2003). Assume $a = a(T, \delta) = (a_1(r), a_2(r), \ldots, a_n(r))$ indicates the ranked score vector. Let the ranked vector r order the genes variables in such a way that $g_1 < g_2 < \ldots < g_n$. Therefore, the log rank score for an observation at T_i is given by

$$a_i = a_i(T,\delta) = \delta_i - \sum_{j=1}^{\gamma_i(T)} \frac{\delta_i}{(n-\gamma_i(T)+1)'}, \tag{4.10}$$

where, $\gamma_i(T) = \sum_{i=1}^{n}\chi\{T_i \leq T_j\}$ is the number of individuals who died or were censored before or at time T_j.

4.3.5 Performance evaluation

We used integrated brier scores (IBS) measure (Graf et al., 1999) to assess and compare the accuracy of the predictive performance of all the models in this study. The IBS represent the average squared differences between the observed survival status and the predicted survival probability at time t. However, the value of the IBS is always between 0 and 1, the value of 0 represent the best possible IBS value. We can calculate the brier scores (BS) measure using the test sample of size n_{test} as follows

$$BS(t) = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \left\{ [0 - \hat{S}(t|x)]^2 \frac{I(t_i \leq t, \delta_i = 1)}{\hat{G}(t_i|x)} + [1 - \hat{S}(t|x)]^2 \frac{I(t_i > t)}{\hat{G}(t|x)} \right\}, \quad (4.11)$$

where $\hat{G}(t|x) \approx P(C > t|X = x)$ is the Kaplan-Meier estimate for the conditional survival function of the censoring times. Therefore, the IBS is calculated as below

$$IBS = \int_0^{max(t)} BS(t)\, dt, \quad (4.12)$$

where $max(t)$ is maximal time for estimating the prediction error curves.

4.4 Results

4.4.1 Cox proportional hazards analysis

The results of the survival problem based on gene expression data were obtained using *R*. We used the Cox PH model based on the selected covariates that satisfy the Cox PH assumptions. We tested the Cox PH assumptions using the Schoenfeld residual test implemented by the function *cox.zhp*. The Cox PH model assumes the regression parameters are constant over time. Therefore, the hazard ratios for any two individuals are constant over time. However, the covariates that do not satisfy the Cox PH assumptions do not meet the criteria to be entered in our final Cox PH

model. As a first step, we fitted the Cox PH model for all the covariates (genes and clinical variables) in our dataset and then obtained the Cox PH assumption using the Schoenfeld residuals Table 4.3. The genes and variables in violation of the Cox PH assumption $(p < 0.05)$ were *DUSP4*, *SYTL1*, and *molecular subtype*.

Table 4.3: Testing the proportional hazard assumption using scaled Schoenfeld residuals.

Probeset ID (Symbol)	χ^{2*} **(df)**	**p-value**
204014_at (DUSP4)	10.219 (1)	0.0014
212947_at (SLC9A8)	1.345 (1)	0.2462
218611_at (IER5)	2.045 (1)	0.1527
219973_at (ARSJ)	3.601 (1)	0.0577
221522_at (ANKRD27)	1.583 (1)	0.2083
221605_s_at (PIPOX)	1.651 (1)	0.1988
227134_at (SYTL1)	4.699 (1)	0.0302
Age at diagnosis (years)	2.589 (1)	0.1076
Molecular subtype	15.824 (5)	0.0074
Disease stages	1.173 (1)	0.2787
Sex	0.378 (1)	0.5388
Tumor location	0.951 (1)	0.3294

* Chi-square statistic.

From the Cox PH model in Table 4.3, three variables violated the Cox PH assumption, and therefore, these genes and *molecular subtype* were not included in the final Cox PH model. We fitted the Cox PH model on the genes and variables that did not violate the Cox PH assumptions before and after imputation. The results from this analysis are shown in Table 4.4. Results before imputation of missing values indicated that 218611_at (*IER5*) (HR=9.51, 95%CI 1.30, 69.58), 221522_at (*ANKRD27*) (HR=34.89, 95%CI 1.91, 635.90), and *late disease stage* (HR=1.97, 95%CI 1.33, 2.93) were associated with higher hazards of death. However, we note that two confidence intervals for *IER5* and *ANKRD27* are quite wide; therefore, they should be interpreted caution. For every year increase, the hazards of death increased by 1.03 (95%CI 1.01, 1.05). Significantly lower hazards

were observed in 212947_at (*SLC9A8*) (HR=0.09, 95%CI 0.02, 0.49), 219973_at (*ARSJ*) (HR=0.23, 95%CI 0.09, 0.58), and 221605_s_at (*PIPOX*) (HR=0.43, 95%CI 0.22, 0.85) differentially expressed genes.

After imputation of missing values, the Cox PH model showed that *sex* was a significant predictor of males having higher death hazards (HR=1.40, 95%CI 1.05, 1.88) than females. Also, the *disease stage* covariate was a significant predictor where those with late disease stage had higher death hazards (HR=1.96, 95%CI 1.47, 2.63) than early cases. Moreover, the results illustrated that 219973_at (*ARSJ*) (HR=0.44, 95%CI 0.22, 0.89), 221605_s_at (*PIPOX*) (HR=0.49, 95%CI 0.28, 0.83) were related with lower hazards of death. For every year increase, the hazards of death increased by 1.03 (95%CI 1.01, 1.04). Significantly higher hazards were detected with gene 218611_at (*IER5*) (HR=6.48, 95%CI 1.37, 30.53) gene.

Table 4.4: Multivariable Cox PH results for predictors of colorectal cancer survival among adults aged 24 years and above.

Probeset ID (Symbol) / Variables	Before imputation (N=307)			After imputation (N=566)		
	HR* (SE)	95%CI	P-value	HR* (SE)	95%CI	P-value
212947_at (SLC9A8)	0.09 (0.84)	(0.02, 0.49)	0.005**	0.30 (0.66)	(0.08, 1.07)	0.066
218611_at (IER5)	9.51 (1.02)	(1.30, 69.58)	0.027*	6.48 (0.79)	(1.37, 30.53)	0.019*
219973_at (ARSJ)	0.23 (0.48)	(0.09, 0.58)	0.002**	0.44 (0.36)	(0.22, 0.89)	0.024*
221522_at (ANKRD27)	34.89 (1.48)	(1.91, 635.90)	0.016*	2.49 (1.06)	(0.31, 19.95)	0.393
221605_s_at (PIPOX)	0.43 (0.34)	(0.22, 0.85)	0.014*	0.49 (0.27)	(0.28, 0.83)	0.009**
Age diagnosis (years)	1.03 (0.01)	(1.01, 1.05)	0.001***	1.03 (0.01)	(1.01, 1.04)	<0.000***
Sex						
Female	1.00			1.00		
Male	1.23 (0.20)	(0.84, 1.81)	0.281	1.40 (0.15)	(1.05, 1.88)	0.024
Stages						
Early	1.00			1.00		
Late	1.97 (0.20)	(1.33, 2.93)	0.001***	1.96 (0.15)	(1.47, 2.63)	<0.000***
Tumor location						
Proximal	1.00			1.00		
Distal	1.06 (0.21)	(0.71, 1.58)	0.783	0.86 (0.16)	(0.63, 1.18)	0.356

HR: Hazard ratio, adjusted for 212947_at, 218611_at, 219973_at, 221522_at, 221605_s_at, age at first diagnosis, sex, disease stage, and tumor location.

4.4.2 Random survival forests analysis

We fitted two random survival forests models, including survival trees built using log-rank and the log-rank-score split rules on the datasets before and after imputation. These two models were built using the 54 genes and the other clinical information as covariates. The characteristics of the two fitted models are summarized in Table 4.5 below.

Table 4.5: Random survival forests results before and after imputation using log-rank and log-rank-score split rules.

	Before imputation (N=246)*		**After imputation (N=453)***	
	Log-rank	**Log-rank-score**	**Log-rank**	**Log-rank-score**
Number of deaths	88	88	157	157
Number of trees	5000	5000	5000	5000
Forest terminal node size	15	15	15	15
Average no. of terminal nodes	13.58	11.92	25.34	22.14
No. of variables tried at each split	8	8	8	8
Total no. of variables	62	62	62	62
Resampling used to grow trees	swor	swor	swor	swor
Resample size used to grow trees	155	155	286	286
Analysis	RSF	RSF	RSF	RSF
Family	surv	surv	surv	surv
Splitting rule	log-rank	log-rank-score	log-rank	log-rank-score
Number of random split points	10	10	10	10
Error rate	41.26%	49.05%	33.22%	43.01%

* Analysis performed using the 80% training set.

Permutation importance measure used to identify the most important genes/ clinical variables associated with the survival of the colon patients (Ehrlinger, 2016; Taylor, 2011; Nasejje & Mwambi, 2017). We fitted a random survival forest model before imputation and after imputation with 5000 survival trees built using log-rank and log-rank-score and their results presented in Figures 4.3 and 4.4.

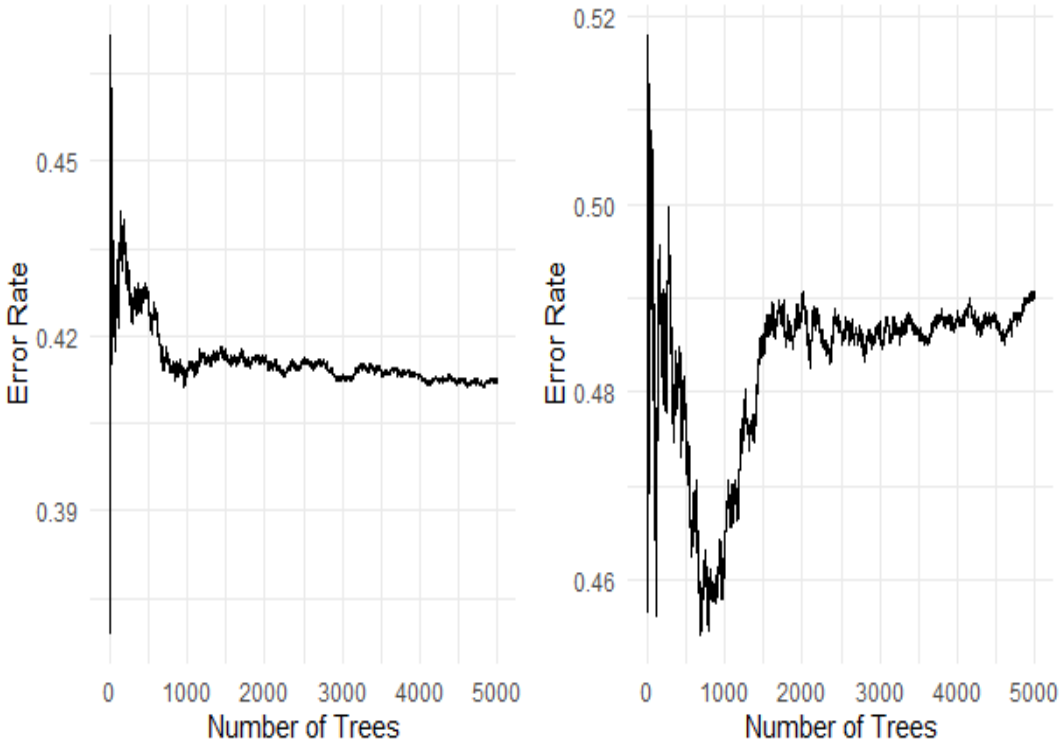


Figure 4.3: The prediction error rate for the random survival forests of 5000 trees before imputation and the log-rank and log-rank-score in the left and right panel used 80% training dataset.

Table 4.5 and Figure 4.3 show that the log-rank split-rule is more stable than the log-rank-score split-rule. Moreover, we fitted the model with 1000, 2000, and 3000 survival trees and noticed that the log-rank-score spilt-rule needs more survival trees to stabilize. In addition, the error rate for the forest built with survival trees based on the log-rank and log-rank-score split-rules are 41.26 and 49.05, respectively. These error rates of the RSF before imputation are much higher than the error rates for RSF built after imputation, as shown in Table 4.5. This result indicates that the imputation can improve the performance of RSF.

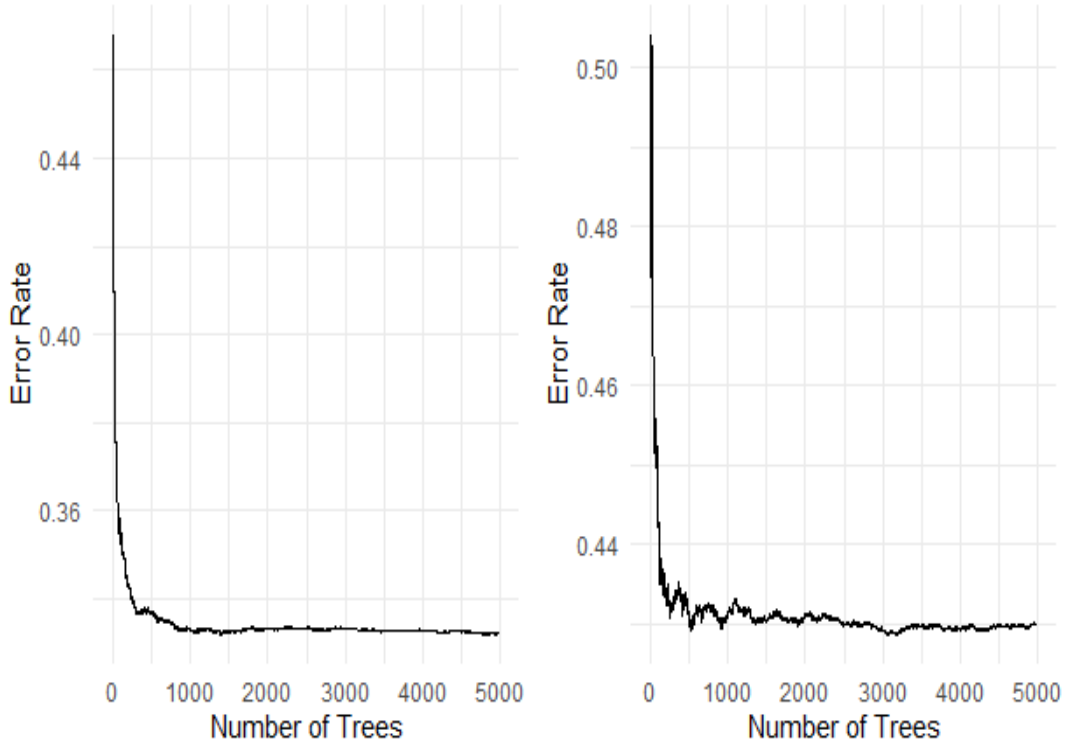


Figure 4.4: The prediction error rate for random survival forests of 5000 trees after imputation and the log-rank and log-rank-score in the left and right panel, respectively, using 80% training dataset.

The genes/ covariates associated with CRC ranked using RSF according to their importance before and after imputation based on the log-rank, and log-rank-score split-rules are presented in Figures 4.5 and 4.6. Using RSF allows all 54 genes and other covariates regardless of their satisfying the Cox PH assumption. However, this is a very important characteristic of the RSF, as explained in the model building stage. The selection of the genes/ covariates in the model does not need to satisfy the too restrictive Cox PH assumption. RSF is purely non-parametric; hence there is no requirement of the Cox PH assumption being satisfied a prior.

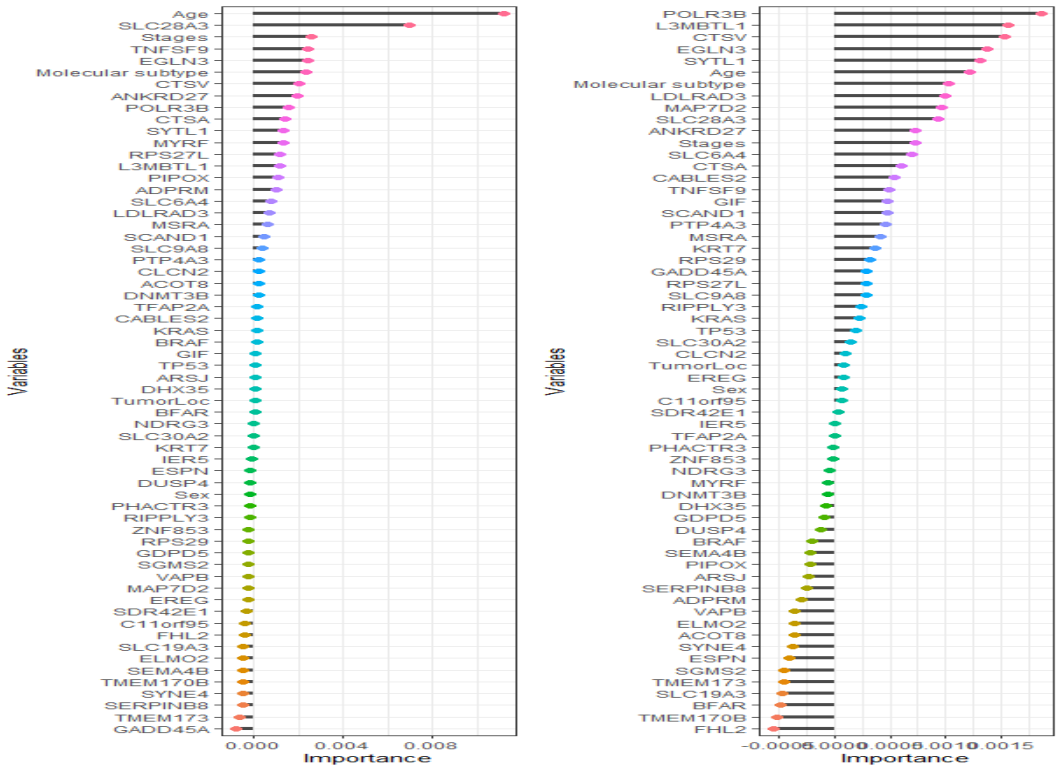


Figure 4.5: The rank of most predictive genes and clinical variables for colorectal cancer patients' survival before the imputation is based on how they influence the survival outcome. The variables importance is built using log-rank and log-rank-score split-rules in the left and right panel, respectively.

We implemented RSF with 5000 survival trees built using two split-rules before and after imputation. The RSF identified the most important genes/ covariates that explain the survival of CRC patients by calculating the measure of the permutation importance as a variable's importance (Ishwaran et al., 2008; Taylor, 2011). For the RSF before imputation see (Figure 4.5), the top 20 genes/ covariates that are most important and strongly associated with the CRC obtained using the log-rank split-rule are *age*, *SLC28A3*, *stages*, *TNFSF9*, *EGLN3*, *molecular subtype*, *CTSV*, *ANKRD27*, *POLR3B*, *CTSA*, *SYTL1*, *MYRF*, *RPS27L*, *L3MBTL1*, *PIPOX*, *ADPRM*, *SLC6A4*, *LDLRAD3*, *MSRA*, and *SCAND1*. While the top 20 genes/ covariates that were identified by RSF using logrank-score are *POLR3B*, *L3MBTL1*, *CTSV*, *EGLN3*, *SYTL1*, *age*, *molecular subtype*, *LDLRAD3*, *MAP7D2*, *SLC28A3*, *ANKRD27*, *stages*, *SLC6A4*, *CTSA*, *CABLES2*, *TNFSF9*, *GIF*, *SCAND1*, *PTP4A3*, and *MSRA*.

However, for the RSF after imputation (Figure 4.6), the top 20 genes/ covariates strongly related to CRC identified using RSF with log-rank split-rule are *age*, *stages*, *molecular subtype*, *PIPOX*, *ADPRM*, *CLCN2*, *RPS27L*, *IER5*, *POLR3B*, *SLC6A4*, *KRAS*, *SGMS2*, *DUSP4*, *SLC28A3*, *SLC9A8*, *ACOT8*, *SYTL1*, *CABLES2*, *SCAND1*, and *MAP7D2*. Although the RSF with logrank-score obtains a top 20 genes/ covariates strongly relevant to CRC, these genes/ covariates are *molecular subtypes*, *POLR3B*, *CLCN2*, *IER5*, *SLC9A8*, *MAP7D2*, *CABLES2*, *SYTL1*, *stages*, *KRAS*, *SLC6A4*, *LDLRAD3*, *CTSA*, *SCAND1*, *PIPOX*, *ARSJ*, *PHACTR3*, *SLC28A3*, *SGMS2*, and *CTSV*.

The RSF with log-rank split-rule after imputation performed better in terms of the error rate. Age and disease stage were the most important covariates that affecting CRC. However, the *PIPOX*, *IER5*, and *SLC9A8* were among the most important genes strongly associated with CRC. These results agree with the results achieved from fitting the Cox PH model presented in Table 4.4. As far as significant effects are concerned, the most striking result to emerge was that the RSF model did pick other genes and covariates as substantial, e.g., *molecular subtype* and *DUSP4* which could not be included in the Cox PH model because of not satisfying the Cox PH assumption.

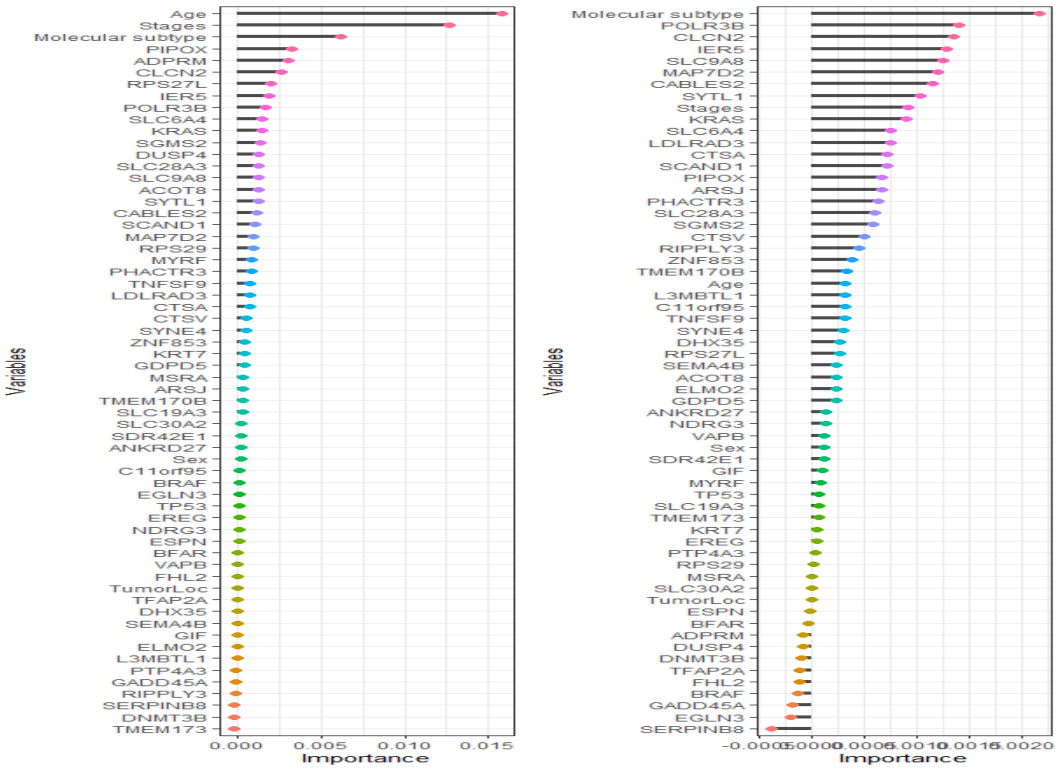


Figure 4.6: The rank of most predictive genes and clinical variables for colorectal cancer patients' survival after the imputation is based on how they influence the survival outcome. The variables importance is built using log-rank and log-rank-score split-rules in the left and right panel, respectively.

4.4.3 Predictive performance

We assessed the predictive performance of the models using the integrated brier scores measure in *R* using the *pec* package (Gerds, 2020; R Core Team, 2021). The model with lower prediction error rates is therefore considered useful (Taylor, 2011; Chen et al., 2011). Figures 4.7 and 4.8 show the prediction error curve of the RSF (log-rank and log-rank score) and Cox PH models before and after imputation. These prediction curves show that Cox PH outperformed RSF with log-rank and log-rank score split rules. The Cox PH model before and after imputation had similar prediction errors, while RSF models under the two split-rules (log-rank and log-rank-score, respectively) after imputation had lower prediction error rates compared to before imputation as can be seen (Figure 4.8). Their predictive performance exhibited that the log-rank split-rule is better than the log-rank-score

split-rule. Moreover, we noticed that the Cox PH model showed good predictive performance compared to the two RSF under the two split-rules before and after imputation models. Thus it is safer to say that if all covariates satisfy the Cox PH assumption, the Cox PH model can be used (Nasejje & Mwambi, 2017).

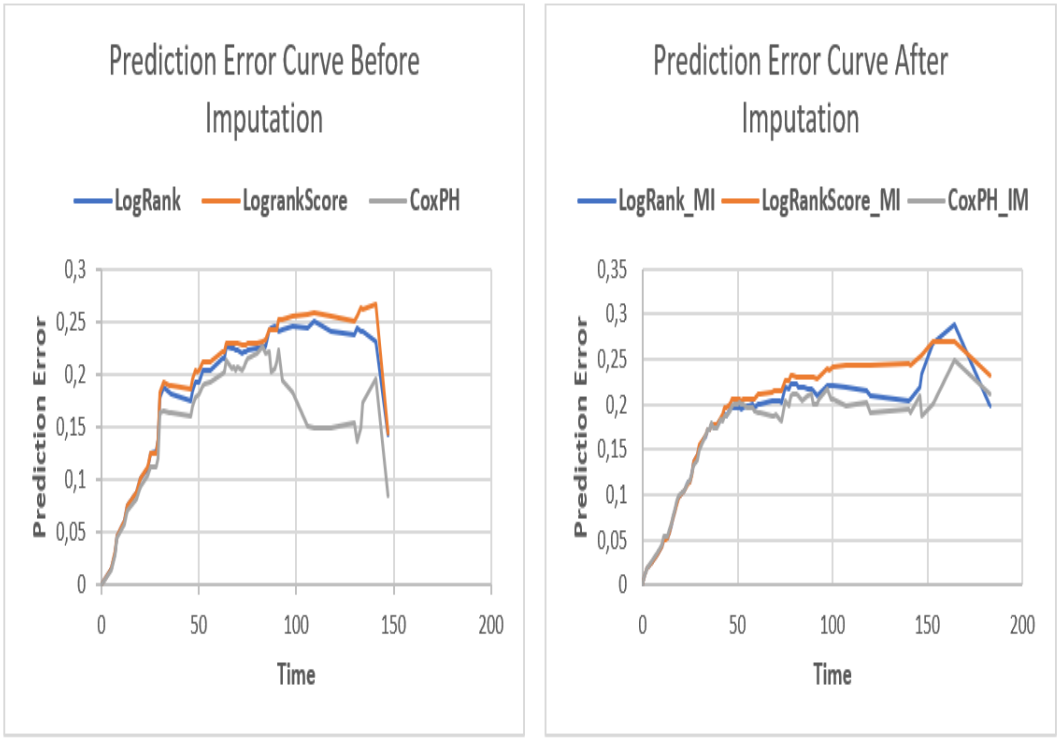


Figure 4.7: RSF with (log-rank and log-rank score) and Cox PH prediction error curve using 20% test set. The complete case and imputed dataset plots are in the left and right panel, respectively.

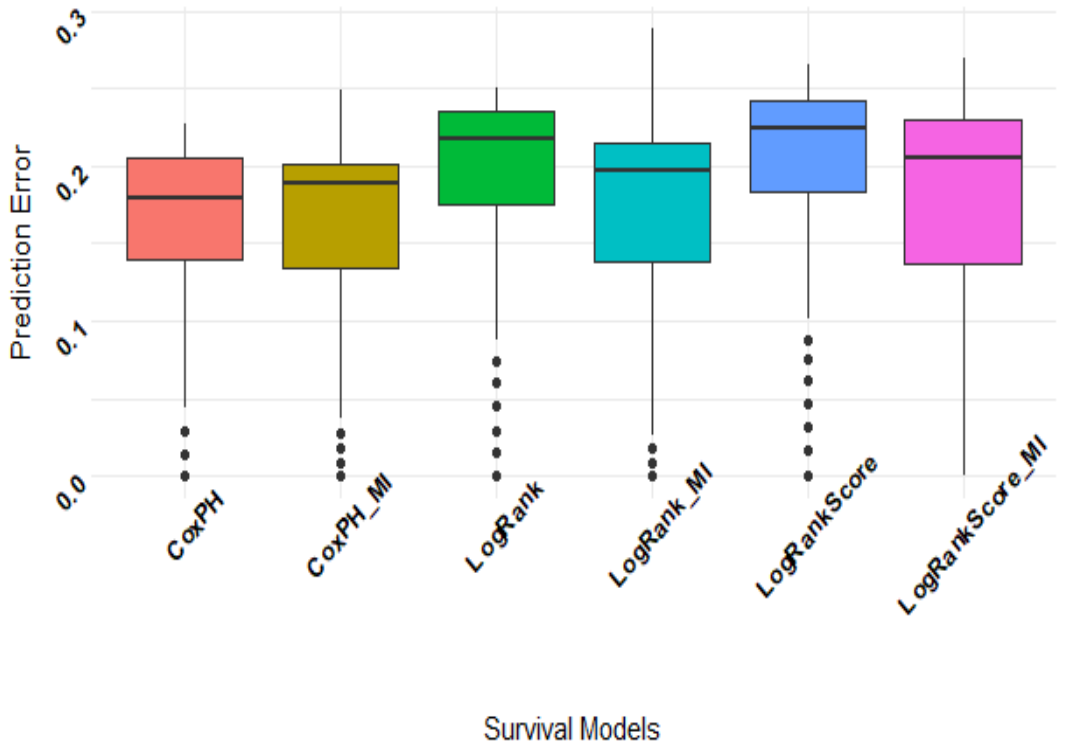


Figure 4.8: RSF with (log-rank and log-rank score) and Cox PH boxplot prediction error using 20% testing set together with the complete case dataset and the imputed data.

Although the Cox PH model before and after imputation had better performance in terms of the prediction error rate, we can still not use it in the event of a violation of the proportionality of hazards assumption. Thus, in the presence of the non-proportional hazards genes/ covariates, using RSF is an appealing option in the analysis of survival data, especially for high dimensional genomics data. Genomics data are usually presented in a matrix, with the columns indicating the samples and the rows showing a genomic feature such as genes (Zang et al., 2016). Table 4.6 shows a comparison of the model performance using the integrated brier scores. We can notice that the prediction error estimates are lower for RSF, especially in the case of using the log-rank as a split rule. In addition, RSF models perform substantially better than Kaplan-Meier and Cox PH models.

Table 4.6: Comparison of the models using the integrated brier scores.

Methods	Before Imputation	After Imputation
Kaplan Meier	0.199	0.201
RSF (Log-rank)	0.192	0.198
RSF (Log-rank score)	0.198	0.202
Cox PH	0.228	0.212

4.5 Discussion

Cancer incidence and mortality are rapidly growing worldwide, exerting big physical, emotional, and financial problems on individual, families, communities, and health systems levels. Cancer is the first or second leading cause of death in 112 countries and is considered the third or fourth in 23 countries (Sung et al., 2021). According to estimates from the World Health Organization (WHO), cancer is the leading cause of death around the world and accounting for nearly 10 million deaths in 2020. Moreover, WHO reported that CRC is the third common new cases, and it is also the second leading cause of death worldwide since 2020 (WHO, 2021b). The study aimed to determine the association between the genes and clinical covariates with CRC survival in the presence of missing values data. We also compared the predictive performance of the Cox PH and RSF models. The study provides essential information for CRC early detection and diagnosis.

The traditional regression-based methods to analyse survival data usually suffer from many problems such as restrictive assumptions including the proportionality, multicollinearity, curse of dimensionality, and lack of ability to rank the predictive performance. However, RSF models are frequently becoming a successful alternative for the analysis of the time to event data. In particular, the RSF is viewed as an appropriate analysing model for survival data, especially when the proportional hazards assumption is violated (Nasejje et al., 2017; Gerds et al., 2013).

When it comes to CRC survival analysis the gene expression and clinical information are utilized as covariates. The gene expression data contains many genes and most of these genes do not discriminate between normal cells and tumors. Therefore, we select the genes in which the change or difference in read counts between two conditions of experiment is statistically significant and such genes are known as the differentially expressed genes. In this study, the differentially expressed genes were obtained using three mutations based on the complete cases. The preliminary analysis showed that 54 potentially differentially expressed genes could be correlated with CRC survival and important for understanding the initiation and progression of CRC. The differentially expressed genes together with the clinical data were used to compare the predictive performance of the Cox PH model and RSF model before and after imputation on the CRC gene expression data.

We used stepwise regression for developing the Cox PH model at a 5% threshold level to get a simple model capturing the association between the top genes and CRC patient survival. Only five genes did not violate the Cox PH assumption in the final Cox PH model. The results show that the error rates of the RSF before imputation are much higher than the error rates for RSF built after imputation. Thus, the imputation can improve the performance of RSF. Although the Cox PH model had a better performance than RSF, the results from the current study demonstrate that the random survival forests models are more flexible than the models based on the Cox PH assumption as a prerequisite for variable inclusion in the model.

After imputation, the Cox PH model indicated *SLC9A8* and *ANKRD27* genes were no longer significant predictors of CRC survival. This because it is expected that the number of observations to increase, hence, statistical power to detect an effect. The variables that were not statistically significant before imputation may now be seen as statistically significant and vice versa. Therefore, this might affect the statistical

power of some variables after imputation. Overall, the most prominent finding to emerge from the analysis based on Cox PH is that for one year increase in age, the hazards of death increase by 1.03, also the males are the most exposed to the hazards of death compared to females. Thus, this study supports evidence from previous observations (van Eeghen et al., 2015; Jiang et al., 2016; Chandrasinghe et al., 2017; White et al., 2018; Abancens et al., 2020).

The results of the RSF using both split-rules before and after imputation identified other genes/ covariates such as *molecular subtype*, *SLC6A4*, *KRAS*, *SGMS2*, *DUSP4*, and *SLC28A3*. These genes/ covariates show up as important in explaining CRC survival rates. However, these genes/ covariates did not appear very strongly associated with CRC survival in the Cox PH model. Thus, one interesting finding to note is that RSF models give additional information about variable importance.

Furthermore, the results from the two RSF models before and after imputation show that *age*, *stages*, *molecular subtype*, *SLC9A8*, *IER5*, *ARSJ*, *ANKRD27*, and *PIPOX* greatly affected the CRC mortality rates. These are ranked in the top 20 variables important in the two RSF models and agree with the Cox PH model results. Contrary to expectations, the RSF model did not pick *sex* as an important variable, while it is significant in the Cox PH model.

The Cox PH model had a better predictive performance in the presence of only those covariates that satisfy the Cox PH assumption compared to the RSF models. This result provides further support for the hypothesis that the Cox PH model works best under this assumption. In contrast, the out-of-bag error rate for the RSF with (log-rank and log-rank-score) before imputation is higher than that after imputation. This result implies that the imputation of missing values is a critical step and enormously improves the model's performance.

The most striking result to emerge from the analysis of the RSF is that log-rank has a better performance compared to the log-rank-score split-rule (Nasejje & Mwambi, 2017). However, with more survival trees the log-rank-score seems to be stabilize compared to a smaller number of survival trees.
We presented the development and validation of a robust five-gene signature (*SLC9A8*, *IER5*, *ARSJ*, *ANKRD27*, and *PIPOX*), which predicted overall survival (OS) for CRC patients. This gene signature was captured using Cox PH and RSF models based on two different scenarios. However, our study results successfully confirmed genes (markers) associated with CRC directly and identified new markers to enrich the field's literature further. Furthermore, the results support previous studies such as Mohammed et al. (Mohammed et al., 2021), where age, sex, and stages were also shown to be related to CRC survival.

4.6 Conclusion

Colorectal cancer (CRC) is a major cause of morbidity and mortality worldwide annually, making CRC the fourth common cause of death from cancer. However, the incidence of CRC has been steadily growing around the world, especially in developing countries. Therefore, the recent advances in technologies such as microarrays allowed for early detection screening using the individual's gene expression profiles.

The present study was designed to identify the genes prognosis of CRC. We developed a robust gene marker associated with the CRC overall survival based on gene expression data generated from microarray, using Cox PH and RSF models before and after missing data imputation. The most prominent finding to emerge from this study is that the Cox PH model identified five genes (*SLC9A8*, *IER5*, *ARSJ*, *ANKRD27*, and *PIPOX*) related to CRC overall survival in addition to *age*, *sex* (after imputation), and *clinical stages*. The RSF model further confirmed these results and had five additional gene markers predicting CRC survival. In addition,

imputation improved the model's performance, and the current findings support the relevance of the missing data imputation. In summary, we recommend using a random survival forests model for survival data, especially in the high dimensional data where many genes might violate the Cox PH assumption.

Chapter 5

General discussion, conclusion, and recommendations

5.1 Discussion of the Main Findings

Non-communicable diseases (NCDs) are a major cause of mortality in developed and developing countries. Among these, cancer which has been documented in the literature to be the first or second leading cause of non-communicable disease mortality before 70 years in 112 of 185 countries. It ranks third in some countries and fourth in other countries (Sung et al., 2021). The burden of cancer incidence and mortality is on the rise in developed and developing countries. The main objective of this study was to integrate a different list of features from both microarray and RNASeq platforms. Also, to propose a model that can predict cancer types and sub-types with high confidence predictions. Moreover, to compare traditional survival models and machine learning models such as RSF.

In Chapter 2, we used CRC gene expression data from both microarray and RNASeq platforms. The DEGs identified from both platforms are based on mutant in comparison to the wild-type mutation status. So, integration is done by combining the different DEGs of the two platforms using intersection, union, and the complement of intersection. Then, we used the combined list of genes to

perform a classification analysis based on the mutation status using statistical and machine learning methods. The methods used are SVM, NB, RF, ANN, KNN, NBLDA, and PLDA. We observed that although microarray and RNASeq platforms might appear different in their data generation, using these gene lists increases the results' reliability. In addition, NBLDA to be the best method when dealing read count RNASeq data. Aslo, survival analysis of the common genes was performed using Cox PH model, which identified 9 genes as prognostic for overall survival in CRC. The important top 5 hub genes that are identified are *ATP6V1C2*, *MYC*, *LMF1*, *HSPA4L*, and *PLAGL2*. Although it is evident that the two platforms do not produce the same information, and the RNASeq platform is considered a replacement for the microarray platform. However, there are a tremendous amount of microarray data that should not be ignored. We conclude that merging data from the two platforms leads to larger data which might help research to obtain more robust statistical significance.

In Chapter 3, we used RNASeq data downloaded from the TCGA database for five common cancers among women. To find an accurate model that predicts cancers types, a stacking ensemble method based on a deep learning model (1D-CNN) is proposed. The stacking approach combines prediction from the base layer classifier to improve the general prediction result. We compared the performance of the proposed stacking ensemble method with the single 1D-CNN as well as with machine learning methods, which include SVM, ANN, KNN, and bagging trees. The results show that the proposed stacking ensemble model to be more promising compared to the single 1-DCNN and machine learning methods. Also, under-sampling is better compared to over-sampling approach. Although the stacking ensemble learning gains accurate prediction, it still has a high computational cost as it demands the training of many similar models. As such high performance computing may be necessary in order to achieve quick results. Overall, we conclude that the stacking ensemble approach will be more reliable than single deep or machine learning models.

In Chapter 4, we used 54 DEGs based on microarray gene expression data download from GEO. We compared traditional and modern survival analysis methods and to find the most predictive genes associated with CRC overall survival. In addition, we compared the performance of the survival methods using the complete cases and imputed data. One interesting finding is that imputation enhances the performance of the methods. In addition, this study found that the RSF model outperformed the Cox PH model possibly because the RSF does not depend on any model assumptions, unlike Cox PH. However further research is required on this aspect including the use of simulation studies which were not the focus of the current work. Moreover, the log-rank split rule is better than the log-rank-score. Overall, the most striking result obtained in this study is that the models identified five genes namely *SLC9A8*, *IER5*, *ARSJ*, *ANKRD27*, *PIPOX*, *sex* (after imputation), and *clinical stages* to be associated to CRC overall survival. This new understanding should help to improve predictions of the impact of these genes/ covariates on CRC.

The NGS technologies have profoundly changed the understanding of diseases such as HIV, TB, and malaria through the gene expression data produced with a higher resolution (Shityakov et al., 2015; Devadas et al., 2016; Tran et al., 2021; Read et al., 2019; Alam et al., 2019; Gebremicael et al., 2019). Therefore, it is possible to hypothesize that the approaches in this study can be implemented and replicated in similar studies of different kinds of diseases. Hence, these approaches will help understand and identify biomarkers that may affect the disease progression and reduce death due to these diseases.

Prior studies that have noted the importance of accurate prediction of cancers early and understanding the genomics characteristics that help improve the treatment policies, such as (Ramirez et al., 2020) used graph CNN (GCNN) to predict tumor and normal samples using gene expression data. Their approach achieved an accuracy between 89.9-94.7%, which is lower than our result in chapter 3. In our study, we have used statistical procedures for gene selection, such as LASSO.

Moreover, we used survival analysis, which identified genes that might help increase the patients' lifestyle and survival rate.

5.2 Study Strengths and Limitations

5.2.1 Study Strengths

We have focused on microarray and RNASeq platforms that provide information about the transcriptome. Moreover, we dealt with the binary and multi-class problems based on gene expression data. In addition, we controlled the imbalanced data problem using resampling techniques, and also we solved the problem of missing data using the multiple imputations technique. Overall, we integrated different sources of data and proposed a new model that provides high performance compared to machine learning methods. Our study identified several genes that are related to different cancers as discussed in chapters 2, 3, and 4. The findings reported here shed new light on the importance of using different data sources, and also on the cancer types and sub-types prediction.

5.2.2 Study Limitations

The major limitation of this study is the small sample size. Also, we did not use of different types of features such as methylation and mutation data, among others. Finally, the study did not evaluate the use of pathways analysis methods, also there is a lack for independent data for validation. It is also proposed that future should consider detailed simulation studies to compare classical statistical models and machine learning methods such as Random Survival Forests and their extensions.

5.3 Conclusion

This thesis has discussed the importance of combining different data sources that proved to strengthen the model performance, compared to a single platform. The stacking approach has promising results compared to single deep and machine

learning methods. This thesis has provided a deeper insight into CRC prediction. Thus, we hope to help the physician in the early detection of CRC, therefore improving the patient's life. Also, the study helps identify a set of genes that will help in determining the tumor types that respond best to a particular therapy. We hope this set of genes will contribute to the knowledge and improve the efficiency, effectiveness, and safety of cancer patients' lives. Finally, combining different methods using stacking approach gains higher prediction performance compare to the single methods.

5.4 Recommendations and Future Research

Considerably more work will need to be done to determine the usefulness of the identified genes in designing therapies for cancer patients. A greater focus on merging different gene expression platforms could produce interesting findings for finding valuable gene markers. Future research should consider integrating more than two platforms, such as microarray, RNASeq, and single-cell RNA sequencing (scRNA-seq).

References

Abancens, M., Bustos, V., Harvey, H., McBryan, J., & Harvey, B. J. (2020). Sexual dimorphism in colon cancer. *Frontiers in oncology*, *10*.

Abusamra, H. (2013). A comparative study of feature selection and classification methods for gene expression data of glioma. *Procedia computer science*, *23*, 5–14.

ACS (2019). Cancer research insights from the latest decade, 2010 to 2020. `https://www.cancer.org/latest-news/cancer-research-insights-from-the-latest-decade-2010-to-2020.html`. Accessed: 2021-07-07.

ACS, A. C. S. (2021). Survival rates for colorectal cancer. `https://www.cancer.org/cancer/colon-rectal-cancer/detection-diagnosis-staging/survival-rates.html`. Accessed: 2021-06-27.

Ajagbe, O. B., Kabair, Z., & O'Connor, T. (2014). Survival analysis of adult tuberculosis disease. *PloS one*, *9*(11), e112838.

Alam, A., Imam, N., Ahmed, M. M., Tazyeen, S., Tamkeen, N., Farooqui, A., Malik, M., Ishrat, R., et al. (2019). Identification and classification of differentially expressed genes and network meta-analysis reveals potential molecular signatures associated with tuberculosis. *Frontiers in genetics*, (p. 932).

AlNuaimi, N., Masud, M. M., Serhani, M. A., & Zaki, N. (2020). Streaming feature selection algorithms for big data: A survey. *Applied computing and informatics*.

Alwan, A., et al. (2011). *Global status report on noncommunicable diseases 2010.*. World Health Organization.

Anders, S., & Huber, W. (2010). Differential expression analysis for sequence count data. *Nature precedings*, (pp. 1–1).

Aryal, S. (2018). Dna microarray. `https://microbenotes.com/dna-microarray/`. Accessed: 2021-09-03.

Auer, P. L., & Doerge, R. W. (2011). A two-stage poisson model for testing rna-seq data. *Statistical applications in genetics and molecular biology*, *10*(1).

Ayesha, S., Hanif, M. K., & Talib, R. (2020). Overview and comparative study of dimensionality reduction techniques for high dimensional data. *Information fusion*, *59*, 44–58.

Aziz, N. A. A., Mokhtar, N. M., Harun, R., Mollah, M. M. H., Rose, I. M., Sagap, I., Tamil, A. M., Ngah, W. Z. W., & Jamal, R. (2016). A 19-gene expression signature as a predictor of survival in colorectal cancer. *BMC medical genomics*, *9*(1), 1–13.

Aziz, R., Verma, C., & 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.

Azur, M. J., Stuart, E. A., Frangakis, C., & Leaf, P. J. (2011). Multiple imputation by chained equations: what is it and how does it work? *International journal of methods in psychiatric research*, *20*(1), 40–49.

Banerjee, A., Chitnis, U., Jadhav, S., Bhawalkar, J., & Chaudhury, S. (2009). Hypothesis testing, type i and type ii errors. *Industrial psychiatry journal*, *18*(2), 127.

Batuwita, R., & Palade, V. (2009). micropred: effective classification of pre-mirnas for human mirna gene prediction. *Bioinformatics*, *25*(8), 989–995.

Bellman, R. (1966). Dynamic programming. *Science*, *153*(3731), 34–37.

Bengio, Y. (2009). *Learning deep architectures for AI*. Now Publishers Inc.

Bengio, Y., Goodfellow, I., & Courville, A. (2017). *Deep learning*, vol. 1. MIT press Massachusetts, USA:.

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.

Benjamini, Y., & Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, (pp. 1165–1188).

Berrar, D. (2019). Cross-validation.

Bian, Q., Chen, J., Qiu, W., Peng, C., Song, M., Sun, X., Liu, Y., Ding, F., Chen, J., & Zhang, L. (2019). Four targeted genes for predicting the prognosis of colorectal cancer: a bioinformatics analysis case. *Oncology letters*, *18*(5), 5043–5054.

Biehl, M. (2019). Supervised learning–an introduction. *Retrieved enero*, *2*, 2021.

Bland, J. M., & Altman, D. G. (1995). Multiple significance tests: the bonferroni method. *British medical journal (Bmj)*, *310*(6973), 170.

Bolstad, B. M., Irizarry, R. A., Åstrand, M., & Speed, T. P. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, *19*(2), 185–193.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on computational learning theory*, (pp. 144–152).

Bradburn, M. J., Clark, T. G., Love, S. B., & Altman, D. G. (2003). Survival analysis part ii: multivariate data analysis–an introduction to concepts and methods. *British journal of cancer*, *89*(3), 431–436.

Bramsen, J. B., Rasmussen, M. H., Ongen, H., Mattesen, T. B., Ørntoft, M.-B. W., Arnadottir, S. S., Sandoval, J., Laguna, T., Vang, S., Øster, B., et al. (2017). Molecular-subtype-specific biomarkers improve prediction of prognosis in colorectal cancer. *Cell reports*, *19*(6), 1268–1280.

Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R. L., Torre, L. A., & Jemal, A. (2018). Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, *68*(6), 394–424.

Breiman, L. (2001a). Random forests. *Machine learning*, *45*(1), 5–32.

Breiman, L. (2001b). Random forests machine learning. 2001; 45: 5–32.

Brown, M., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C., Ares, M., & Haussler, D. (1999). Support vector machine classification of microarray gene expression data. *University of california, santa cruz, technical Report UCSC-CRL-99-09*.

Browner, W. S., Newman, T. B., Cummings, S., & Hulley, S. (2001). Getting ready to estimate sample size: hypotheses and underlying principles. *Designing clinical research: An epidemiologic approach. 2nd ed. Philadelphia: Lippincott Williams and Wilkins*, (p. 56).

Bühlmann, P., & Van De Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media.

Bullard, J. H., Purdom, E., Hansen, K. D., & Dudoit, S. (2010). Evaluation of statistical methods for normalization and differential expression in mrna-seq experiments. *BMC bioinformatics*, *11*(1), 1–13.

Carpenter, J., & Kenward, M. (2012). *Multiple imputation and its application*. John Wiley & Sons.

Castillo, D., Galvez, J. M., Herrera, L. J., Rojas, F., Valenzuela, O., Caba, O., Prados, J., & Rojas, I. (2019). Leukemia multiclass assessment and classification from

microarray and rna-seq technologies integration at gene expression level. *PloS one*, *14*(2), e0212127.

Castillo, D., Gálvez, J. M., Herrera, L. J., San Román, B., Rojas, F., & Rojas, I. (2017). Integration of rna-seq data with heterogeneous microarray data for breast cancer profiling. *BMC bioinformatics*, *18*(1), 1–15.

Cerami, E., Gao, J., Dogrusoz, U., Gross, B. E., Sumer, S. O., Aksoy, B. A., Jacobsen, A., Byrne, C. J., Heuer, M. L., Larsson, E., et al. (2012). The cbio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data.

Chaba, L. (2016). Evaluation of methods for gene selection in melanoma cell lines. *International journal of statistics in medical research*.

Chandrasinghe, P. C., Ediriweera, D. S., Nazar, T., Kumarage, S., Hewavisenthi, J., & Deen, K. I. (2017). Overall survival of elderly patients having surgery for colorectal cancer is comparable to younger patients: results from a south asian population. *Gastroenterology research and practice*, *2017*.

Chang, K., Creighton, C., Davis, C., Donehower, L., et al. (2013). The cancer genome atlas pan-cancer analysis project. *Nature genetics*, *45*(10), 1113–1120.

Chatzimparmpas, A., Martins, R. M., Kucher, K., & Kerren, A. (2020). Stackgenvis: Alignment of data, algorithms, and models for stacking ensemble learning using performance metrics. *IEEE transactions on visualization and computer graphics*, *27*(2), 1547–1557.

Chawla, N. V. (2009). Data mining for imbalanced datasets: An overview. *Data mining and knowledge discovery handbook*, (pp. 875–886).

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Chen, G., Kim, S., Taylor, J. M., Wang, Z., Lee, O., Ramnath, N., Reddy, R. M., Lin, J., Chang, A. C., Orringer, M. B., et al. (2011). Development and validation

of a quantitative real-time polymerase chain reaction classifier for lung cancer prognosis. *Journal of thoracic oncology*, *6*(9), 1481–1487.

Chen, X., & Ishwaran, H. (2012). Random forests for genomic data analysis. *Genomics*, *99*(6), 323–329.

Chernick, M. R., González-Manteiga, W., Crujeiras, R. M., & Barrios, E. B. (2011). Bootstrap methods.

Chin, L., Andersen, J. N., & Futreal, P. A. (2011a). Cancer genomics: from discovery science to personalized medicine. *Nature medicine*, *17*(3), 297–303.

Chin, L., Hahn, W. C., Getz, G., & Meyerson, M. (2011b). Making sense of cancer genomic data. *Genes & development*, *25*(6), 534–555.

Chu, F., & Wang, L. (2003). Gene expression data analysis using support vector machines. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, vol. 3, (pp. 2268–2271). IEEE.

Churko, J. M., Mantalas, G. L., Snyder, M. P., & Wu, J. C. (2013). Overview of high throughput sequencing technologies to elucidate molecular pathways in cardiovascular diseases. *Circulation research*, *112*(12), 1613–1623.

Ciampi, A., Chang, C.-H., Hogg, S., & McKinney, S. (1987). Recursive partition: A versatile method for exploratory-data analysis in biostatistics. In *Biostatistics*, (pp. 23–50). Springer.

Ciregan, D., Meier, U., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *2012 IEEE conference on computer vision and pattern recognition*, (pp. 3642–3649). IEEE.

Colaprico, A., Silva, T. C., Olsen, C., Garofano, L., Cava, C., Garolini, D., Sabedot, T. S., Malta, T. M., Pagnotta, S. M., Castiglioni, I., et al. (2016). Tcgabiolinks: an r/bioconductor package for integrative analysis of tcga data. *Nucleic acids research*, *44*(8), e71–e71.

Colombo, P.-E., Milanezi, F., Weigelt, B., & Reis-Filho, J. S. (2011). Microarrays in the 2010s: the contribution of microarray-based gene expression profiling to breast cancer classification, prognostication and prediction. *Breast cancer research*, *13*(3), 1–15.

Crow, M., Lim, N., Ballouz, S., Pavlidis, P., & Gillis, J. (2019). Predictability of human differential gene expression. *Proceedings of the national academy of sciences*, *116*(13), 6491–6500.

Cunningham, P., Cord, M., & Delany, S. J. (2008). Supervised learning. In *Machine learning techniques for multimedia*, (pp. 21–49). Springer.

Cunningham, P., & Delany, S. J. (2020). k-nearest neighbour classifiers: (with python examples). *arXiv preprint arXiv:2004.04523*.

Dai, G.-P., Wang, L.-P., WEn, Y.-Q., REn, X.-Q., & Zuo, S.-G. (2020). Identification of key genes for predicting colorectal cancer prognosis by integrated bioinformatics analysis. *Oncology letters*, *19*(1), 388–398.

Datta, S., & Nettleton, D. (2014). *Statistical analysis of next generation sequencing data*. Springer.

De Campos, L. M., Cano, A., Castellano, J. G., & Moral, S. (2011). Bayesian networks classifiers for gene-expression data. In *2011 11th International Conference on Intelligent Systems Design and Applications*, (pp. 1200–1206). IEEE.

Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and trends in signal processing*, *7*(3–4), 197–387.

DeSantis, C. E., Miller, K. D., Goding Sauer, A., Jemal, A., & Siegel, R. L. (2019). Cancer statistics for african americans, 2019. *CA: a cancer journal for clinicians*, *69*(3), 211–233.

Devadas, K., Biswas, S., Haleyurgirisetty, M., Wood, O., Ragupathy, V., Lee, S., & Hewlett, I. (2016). Analysis of host gene expression profile in hiv-1 and hiv-2 infected t-cells. *PloS one*, *11*(1), e0147421.

Di, Y., Schafer, D. W., Cumbie, J. S., & Chang, J. H. (2011). The nbp negative binomial model for assessing differential gene expression from rna-seq. *Statistical applications in genetics and molecular biology*, *10*(1).

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, (pp. 1–15). Springer.

Dillies, M.-A., Rau, A., Aubert, J., Hennequet-Antier, C., Jeanmougin, M., Servant, N., Keime, C., Marot, G., Castel, D., Estelle, J., et al. (2013). A comprehensive evaluation of normalization methods for illumina high-throughput rna sequencing data analysis. *Briefings in bioinformatics*, *14*(6), 671–683.

Dixon, P. M. (2006). Bootstrap resampling. *Encyclopedia of environmetrics*, *1*.

Dong, K., Zhao, H., Tong, T., & Wan, X. (2016). Nblda: negative binomial linear discriminant analysis for rna-seq data. *BMC bioinformatics*, *17*(1), 369.

Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2020). A survey on ensemble learning. *Frontiers of computer science*, *14*(2), 241–258.

Dong, Y., & Peng, C.-Y. J. (2013). Principled missing data methods for researchers. *SpringerPlus*.

Dry, J. R., Pavey, S., Pratilas, C. A., Harbron, C., Runswick, S., Hodgson, D., Chresta, C., McCormack, R., Byrne, N., Cockerill, M., et al. (2010). Transcriptional pathway signatures predict mek addiction and response to selumetinib (azd6244). *Cancer research*, *70*(6), 2264–2273.

Dündar, F., Skrabanek, L., & Zumbo, P. (2015). Introduction to differential gene expression analysis using rna-seq. *Applied bioinformatics core/weill cornell medical college*, (pp. 1–67).

Dwivedi, A. K. (2018). Artificial neural network model for effective cancer classification using microarray gene expression data. *Neural computing and applications*, *29*(12), 1545–1554.

Ehrlinger, J. (2016). ggrandomforests: Exploring random forest survival. *arXiv preprint arXiv:1612.08974*.

El Naqa, I., & Murphy, M. J. (2015). What is machine learning? In *machine learning in radiation oncology*, (pp. 3–11). Springer.

Elbashir, M. K., Ezz, M., Mohammed, M., & Saloum, S. S. (2019). Lightweight convolutional neural network for breast cancer classification using rna-seq gene expression data. *IEEE Access*, *7*, 185338–185348.

Farswan, A., Gupta, A., Gupta, R., & Kaur, G. (2020). Imputation of gene expression data in blood cancer and its significance in inferring biological pathways. *Frontiers in oncology*, *9*, 1442.

Favoriti, P., Carbone, G., Greco, M., Pirozzi, F., Pirozzi, R. E. M., & Corcione, F. (2016). Worldwide burden of colorectal cancer: a review. *Updates in surgery*, *68*(1), 7–11.

Fonti, V., & Belitser, E. (2017). Feature selection using lasso. *VU Amsterdam research paper in business analytics*, *30*, 1–25.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, *33*(1), 1.

Friedman, J., Hastie, T., Tibshirani, R., et al. (2001). *The elements of statistical learning*, vol. 1. Springer series in statistics New York.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2011). A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)*, *42*(4), 463–484.

Gandomani, H. S., Aghajani, M., Mohammadian-Hafshejani, A., Tarazoj, A. A., Pouyesh, V., Salehiniya, H., et al. (2017). Colorectal cancer in the world: incidence, mortality and risk factors. *Biomedical research and therapy*, *4*(10), 1656–1675.

García-Díaz, P., Sánchez-Berriel, I., Martínez-Rojas, J. A., & Diez-Pascual, A. M. (2020). Unsupervised feature selection algorithm for multiclass cancer classification of gene expression rna-seq data. *Genomics*, *112*(2), 1916–1925.

Gebremicael, G., Kassa, D., Alemayehu, Y., Gebreegziaxier, A., Kassahun, Y., van Baarle, D., HM Ottenhoff, T., M. Cliff, J., & C. Haks, M. (2019). Gene expression profiles classifying clinical stages of tuberculosis and monitoring treatment responses in ethiopian hiv-negative and hiv-positive cohorts. *PloS one*, *14*(12), e0226137.

Gerds, T. A. (2020). *Package 'pec'*.

Gerds, T. A., Kattan, M. W., Schumacher, M., & Yu, C. (2013). Estimating a time-dependent concordance index for survival prediction models with covariate dependent censoring. *Statistics in medicine*, *32*(13), 2173–2184.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, *286*(5439), 531–537.

Gomez-Cabrero, D., Abugessaisa, I., Maier, D., Teschendorff, A., Merkenschlager, M., Gisel, A., Ballestar, E., Bongcam-Rudloff, E., Conesa, A., & Tegnér, J. (2014). Data integration in the era of omics: current and future challenges. *BMC systems biology*, *8*(2), 1–10.

Govindarajan, R., Duraiyan, J., Kaliyappan, K., & Palanisamy, M. (2012). Microarray and its applications. *Journal of pharmacy & bioallied sciences*, *4*(Suppl 2), S310.

Graf, E., Schmoor, C., Sauerbrei, W., & Schumacher, M. (1999). Assessment and comparison of prognostic classification schemes for survival data. *Statistics in medicine*, *18*(17-18), 2529–2545.

Granados-Romero, J. J., Valderrama-Treviño, A. I., Contreras-Flores, E. H., Barrera-Mera, B., Herrera Enríquez, M., Uriarte-Ruíz, K., Ceballos-Villalba, J., Estrada-Mata, A. G., Alvarado Rodríguez, C., & Arauz-Peña, G. (2017). Colorectal cancer: a review. *International journal of research in medical sciences*, *5*(11), 4667–4676.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, *3*(Mar), 1157–1182.

Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. A. (2008). *Feature extraction: foundations and applications*, vol. 207. Springer.

Halpin, H. A., Morales-Suárez-Varela, M. M., & Martin-Moreno, J. M. (2010). Chronic disease prevention and the new public health. *Public health reviews*, *32*(1), 120–154.

Harbers, M., & Kahl, G. (2011). *Tag-based next generation sequencing*. John Wiley & Sons.

Hart, P. E., Stork, D. G., & Duda, R. O. (2000). *Pattern classification*. Wiley Hoboken.

Hasan, A., & Adnan, M. A. (2012). High dimensional microarray data classification using correlation based feature selection. In *2012 International conference on biomedical engineering (ICoBE)*, (pp. 319–321). IEEE.

Hastie, T., Qian, J., & Tay, K. (2016). An introduction to glmnet.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). The elements of statistical learning. springer series in statistics. In :. Springer.

Haury, A.-C., Gestraud, P., & Vert, J.-P. (2011). The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. *PloS one*, *6*(12), e28210.

He, W.-L., Weng, X.-T., Wang, J.-L., Lin, Y.-K., Liu, T.-W., Zhou, Q.-Y., Hu, Y., Pan, Y., & Chen, X.-L. (2018). Association between c-myc and colorectal cancer prognosis: a meta-analysis. *Frontiers in physiology*, *9*, 1549.

Heaton, J. (2020). Applications of deep neural networks. *arXiv preprint arXiv:2009.05673*.

Herzog, M. H., Francis, G., & Clarke, A. (2019). The multiple testing problem. In *Understanding statistics and experimental design*, (pp. 63–66). Springer.

Hoens, T. R., & Chawla, N. V. (2013). Imbalanced datasets: from sampling to classifiers. *Imbalanced learning: Foundations, algorithms, and applications*, (pp. 43–59).

Hordri, N. F., Yuhaniz, S. S., & Shamsuddin, S. M. (2016). Deep learning and its applications: a review. In *Conference on Postgraduate Annual Research on Informatics Seminar*.

Hothorn, T., & Lausen, B. (2003). On the exact distribution of maximally selected rank statistics. *Computational statistics & data analysis*, *43*(2), 121–137.

Hu, H., Li, J., Plank, A., Wang, H., & Daggard, G. (2006). A comparative study of classification methods for microarray data analysis. In *Proceedings of the 5th Australasian data mining conference (AusDM 2006): Data mining and analytics 2006*, (pp. 33–37). ACS Press.

Huang, D. W., Sherman, B. T., Tan, Q., Kir, J., Liu, D., Bryant, D., Guo, Y., Stephens, R., Baseler, M. W., Lane, H. C., et al. (2007). David bioinformatics resources: expanded annotation database and novel algorithms to better extract biology from large gene lists. *Nucleic acids research*, *35*(suppl_2), W169–W175.

Huque, M. H., Carlin, J. B., Simpson, J. A., & Lee, K. J. (2018). A comparison of multiple imputation methods for missing data in longitudinal studies. *BMC medical research methodology*, *18*(1), 1–16.

Hurwitz, J., & Kirsch, D. (2018). Machine learning for dummies. *IBM Limited Edition*, *75*.

Inza, I., Larranaga, P., Blanco, R., & Cerrolaza, A. J. (2004). Filter versus wrapper gene selection approaches in dna microarray domains. *Artificial intelligence in medicine*, *31*(2), 91–103.

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., & Lauer, M. S. (2008). Random survival forests. *The annals of applied statistics*, *2*(3), 841–860.

Ishwaran, H., Kogalur, U. B., & Kogalur, M. U. B. (2021). Package 'randomforestsrc'.

Jain, I., Jain, V. K., & Jain, R. (2018). Correlation feature selection based improved-binary particle swarm optimization for gene selection and cancer classification. *Applied soft computing*, *62*, 203–215.

Jakobsen, J. C., Gluud, C., Wetterslev, J., & Winkel, P. (2017). When and how should multiple imputation be used for handling missing data in randomised clinical trials–a practical guide with flowcharts. *BMC medical research methodology*, *17*(1), 1–10.

Jaluria, P., Konstantopoulos, K., Betenbaugh, M., & Shiloach, J. (2007). A perspective on microarrays: current applications, pitfalls, and potential uses. *Microbial cell factories*, *6*(1), 1–14.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013a). *An introduction to statistical learning*, vol. 112. Springer.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013b). Resampling methods. In *An introduction to statistical learning*, (pp. 175–201). Springer.

Jemal, A., Bray, F., Center, M. M., Ferlay, J., Ward, E., & Forman, D. (2011). Global cancer statistics. *CA: a cancer journal for clinicians*, *61*(2), 69–90.

Jiang, Z., Wang, X., Tan, X., & Fan, Z. (2016). Effect of age on survival outcome in operated and non-operated patients with colon cancer: a population-based study. *PLoS One*, *11*(1), e0147383.

Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of big data*, *6*(1), 1–54.

Jović, A., Brkić, K., & Bogunović, N. (2015). A review of feature selection methods with applications. In *2015 38th international convention on information and*

communication technology, electronics and microelectronics (MIPRO), (pp. 1200–1205). Ieee.

Kang, H. (2013). The prevention and handling of the missing data. *Korean J Anesthesiol*, *64*(5), 402–406.

Karatzoglou, A., Smola, A., Hornik, K., & Karatzoglou, M. A. (2019). Package 'kernlab'. *CRAN R Project*.

Kim, T. K. (2015). T test as a parametric statistic. *Korean journal of anesthesiology*, *68*(6), 540.

Kira, K., & Rendell, L. A. (1992). A practical approach to feature selection. In *Machine learning proceedings 1992*, (pp. 249–256). Elsevier.

Kleinbaum, D. G., & Klein, M. (2010). *Survival analysis*, vol. 3. Springer.

Koch, C. M., Chiu, S. F., Akbarpour, M., Bharat, A., Ridge, K. M., Bartom, E. T., & Winter, D. R. (2018). A beginner's guide to analysis of rna sequencing data. *American journal of respiratory cell and molecular biology*, *59*(2), 145–157.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence*, *97*(1-2), 273–324.

Kononenko, I. (2001). Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine*, *23*(1), 89–109.

Kotsiantis, S., Kanellopoulos, D., Pintelas, P., et al. (2006). Handling imbalanced datasets: A review. *GESTS international transactions on computer science and engineering*, *30*(1), 25–36.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*, 1097–1105.

Kuhn, M., Wing, J., Weston, S., Williams, A., Keefer, C., Engelhardt, A., Cooper, T., Mayer, Z., Kenkel, B., Team, R. C., et al. (2020). Package 'caret'. *The R journal*, *223*.

Kulski, J. (2016). *Next generation sequencing: advances, applications and challenges*. BoD–Books on Demand.

Kuo, F. Y., & Sloan, I. H. (2005). Lifting the curse of dimensionality. *Notices of the AMS*, *52*(11), 1320–1328.

Kwon, Y. M., & Ricke, S. C. (2011). *High-throughput next generation sequencing: methods and applications*. Springer.

Lai, Y. (2010). Differential expression analysis of digital gene expression data: Rna-tag filtering, comparison of t-type tests and their genome-wide co-expression based adjustments. *International journal of bioinformatics research and applications*, *6*(4), 353–365.

Landau, W. M., & Liu, P. (2013). Dispersion estimation and its effect on test performance in rna-seq data analysis: a simulation-based comparison of methods. *PloS one*, *8*(12), e81415.

Law, C. W., Chen, Y., Shi, W., & Smyth, G. K. (2014). voom: Precision weights unlock linear model analysis tools for rna-seq read counts. *Genome biology*, *15*(2), 1–17.

Lazar, C., Taminau, J., Meganck, S., Steenhoff, D., Coletta, A., Molter, C., de Schaetzen, V., Duque, R., Bersini, H., & Nowe, A. (2012). A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE/ACM transactions on computational biology and bioinformatics*, *9*(4), 1106–1119.

Learned-Miller, E. G. (2014). Introduction to supervised learning. *I: Department of computer science, University of Massachusetts*.

Lee, K., Jeong, H.-o., Lee, S., & Jeong, W.-K. (2019). Cpem: Accurate cancer type classification based on somatic alterations using an ensemble of a random forest and a deep neural network. *Scientific reports*, *9*(1), 1–9.

Lek, S., & Park, Y. (2008). Artificial neural networks. In *Encyclopedia of Ecology, Five-Volume Set*, (pp. 237–245). Elsevier Inc.

Leung, M. K., Delong, A., Alipanahi, B., & Frey, B. J. (2015). Machine learning in genomic medicine: a review of computational problems and data sets. *Proceedings of the IEEE*, *104*(1), 176–197.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. *ACM computing surveys (CSUR)*, *50*(6), 1–45.

Li, W. (2012). Volcano plots in analyzing differential expressions with mrna microarrays. *Journal of bioinformatics and computational biology*, *10*(06), 1231003.

Liu, B. (2011). Supervised learning. In *Web data mining*, (pp. 63–132). Springer.

Liu, H., Gu, X., Wang, G., Huang, Y., Ju, S., Huang, J., & Wang, X. (2019). Copy number variations primed lncrnas deregulation contribute to poor prognosis in colorectal cancer. *Aging (Albany NY)*, *11*(16), 6089.

Love, M. I., Anders, S., Kim, V., & Huber, W. (2019). Rna-seq workflow: gene-level exploratory analysis and differential expression. `https://bioconductor.org/packages/release/workflows/vignettes/rnaseqGene/inst/doc/rnaseqGene.html#differential-expression-analysis`. Accessed: 2020-05-01.

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.

Lowe, R., Shirley, N., Bleackley, M., Dolan, S., & Shafee, T. (2017). Transcriptomics technologies. *PLoS computational biology*, *13*(5), e1005457.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, (pp. 4768–4777).

Lusa, L., et al. (2010). Class prediction for high-dimensional class-imbalanced data. *BMC bioinformatics*, *11*(1), 1–17.

Magoulas, G. D., & Prentza, A. (1999). Machine learning in medical applications. In *Advanced course on artificial intelligence*, (pp. 300–307). Springer.

Mamatjan, Y., Agnihotri, S., Goldenberg, A., Tonge, P., Mansouri, S., Zadeh, G., & Aldape, K. (2017). Molecular signatures for tumor classification: an analysis of the cancer genome atlas data. *The journal of molecular diagnostics*, *19*(6), 881–891.

Marco-Puche, G., Lois, S., Benítez, J., & Trivino, J. C. (2019). Rna-seq perspectives to improve clinical diagnosis. *Frontiers in genetics*, *10*, 1152.

Marisa, L., de Reyniès, A., Duval, A., Selves, J., Gaub, M. P., Vescovo, L., Etienne-Grimaldi, M.-C., Schiappa, R., Guenot, D., Ayadi, M., et al. (2013). Gene expression classification of colon cancer into molecular subtypes: characterization, validation, and prognostic value. *PLoS medicine*, *10*(5), e1001453.

Mármol, I., Sánchez-de Diego, C., Pradilla Dieste, A., Cerrada, E., & Rodriguez Yoldi, M. J. (2017). Colorectal carcinoma: a general overview and future perspectives in colorectal cancer. *International journal of molecular sciences*, *18*(1), 197.

Martinez-Romero, J., Bueno-Fortes, S., Martín-Merino, M., de Molina, A. R., & De Las Rivas, J. (2018). Survival marker genes of colorectal cancer derived from consistent transcriptomic profiling. *BMC genomics*, *19*(8), 45–60.

Massey, A., & Miller, S. J. (2006). Tests of hypotheses using statistics. *Mathematics department, Brown University, Providence, RI*, *2912*, 1–32.

Mathew, A., Amudha, P., & Sivakumari, S. (2020). Deep learning techniques: An overview. In *International Conference on Advanced Machine Learning Technologies and Applications*, (pp. 599–608). Springer.

May, F. P., & Anandasabapathy, S. (2019). Colon cancer in africa: Primetime for screening? *Gastrointestinal endoscopy*, *89*(6), 1238–1240.

Mboya, I. B., Mahande, M. J., Obure, J., & Mwambi, H. G. (2020). Predictors of perinatal death in the presence of missing data: A birth registry-based study in northern tanzania. *Plos one*, *15*(4), e0231636.

Mboya, I. B., Mahande, M. J., Obure, J., & Mwambi, H. G. (2021). Predictors of singleton preterm birth using multinomial regression models accounting for missing data: A birth registry-based cohort study in northern tanzania. *Plos one*, *16*(4), e0249411.

Mlambo, N., Cheruiyot, W. K., & Kimwele, M. W. (2016). A survey and comparative study of filter and wrapper feature selection techniques. *International journal of engineering and science (IJES)*, *5*(8), 57–67.

Moguerza, J. M., & Muñoz, A. (2006). Support vector machines with applications. *Statistical science*, *21*(3), 322–336.

Mohammed, M., Mwambi, H., & Omolo, B. (2021). Colorectal cancer classification and survival analysis based on an integrated rna and dna molecular signature. *Current bioinformatics*, *16*(4), 583–600.

Mohammed, M., Mwambi, H., Omolo, B., & Elbashir, M. K. (2018). Using stacking ensemble for microarray-based cancer classification. In *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, (pp. 1–8). IEEE.

Mohammed, M. M. A. (2018). *A comparison of cancer classification methods based on microarray data*. Master's thesis, School of Mathematics, Statistics and Computer Science, King Edward Ave, Scottsville, Pietermaritzburg, South Africa, 3209.

Molenberghs, G., & Kenward, M. (2007). *Missing data in clinical studies*, vol. 61. John Wiley & Sons.

Molinari, C., Marisi, G., Passardi, A., Matteucci, L., De Maio, G., & Ulivi, P. (2018). Heterogeneity in colorectal cancer: a challenge for personalized medicine? *International journal of molecular sciences*, *19*(12), 3733.

Morhason-Bello, I. O., Odedina, F., Rebbeck, T. R., Harford, J., Dangou, J.-M., Denny, L., & Adewole, I. F. (2013). Challenges and opportunities in cancer control in africa: a perspective from the african organisation for research and training in cancer. *The lancet oncology*, *14*(4), e142–e151.

Mostavi, M., Chiu, Y.-C., Huang, Y., & Chen, Y. (2020). Convolutional neural network models for cancer type prediction based on gene expression. *BMC medical genomics*, *13*(5), 1–13.

Moten, A., Schafer, D., & Ferrari, M. (2014). Redefining global health priorities: Improving cancer care in developing settings. *Journal of global health*, *4*(1).

Mulder, N., Adebamowo, C. A., Adebamowo, S. N., Adebayo, O., Adeleye, O., Alibi, M., Baichoo, S., Benkahla, A., Fadlelmola, F. M., Ghazal, H., et al. (2017). Genomic research data generation, analysis and sharing–challenges in the african setting. *Data science journal*, *16*.

Munoz, A., de Diego, I. M., & Moguerza, J. M. (2003). Support vector machine classifiers for asymmetric proximities. In *Artificial Neural Networks and Neural Information Processing—ICANN/ICONIP 2003*, (pp. 217–224). Springer.

Myte, R. (2013). Covariate selection for colorectal cancer survival data: A comparison case study between random survival forests and the cox proportional-hazards model.

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.

Nasejje, J. B., & Mwambi, H. (2017). Application of random survival forests in understanding the determinants of under-five child mortality in uganda in the presence of covariates that satisfy the proportional and non-proportional hazards assumption. *BMC research notes*, *10*(1), 1–18.

Nasejje, J. B., Mwambi, H., Dheda, K., & Lesosky, M. (2017). A comparison of the conditional inference survival forest model to random survival forests based on a simulation study as well as on two applications with time-to-event data. *BMC medical research methodology*, *17*(1), 1–17.

Nawaz, M., Sewissy, A. A., & Soliman, T. H. A. (2018). Multi-class breast cancer classification using deep learning convolutional neural network. *International journal of advanced computer science and applications*, *9*(6), 316–332.

Nelson, W. (1972). Theory and applications of hazard plotting for censored failure data. *Technometrics*, *14*(4), 945–966.

Nguyen, T. N. H. (2020). *Combining machine learning and reference-free transcriptome analysis for the identification of prostate cancer signatures*. Ph.D. thesis, Université Paris-Saclay.

Ogino, S., Nowak, J. A., Hamada, T., Phipps, A. I., Peters, U., Milner Jr, D. A., Giovannucci, E. L., Nishihara, R., Giannakis, M., Garrett, W. S., et al. (2018). Integrative analysis of exogenous, endogenous, tumour and immune factors for precision medicine. *Gut*, *67*(6), 1168–1180.

Ogutu, J. O., Schulz-Streeck, T., & Piepho, H.-P. (2012). Genomic selection using regularized linear regression models: ridge regression, lasso, elastic net and their extensions. In *BMC proceedings*, vol. 6, (pp. 1–6). Springer.

Olsen, M. (2015). *Cancer in Sub-Saharan Africa: the need for new paradigms in global health*. Boston University Frederick S. Pardee Center for the Study of the Longer ....

Omolo, B., Yang, M., Lo, F. Y., Schell, M. J., Austin, S., Howard, K., Madan, A., & Yeatman, T. J. (2016). Adaptation of a ras pathway activation signature from ff to ffpe tissues in colorectal cancer. *BMC medical genomics*, *9*(1), 1–10.

Organization, W. H. (2002). *National cancer control programmes: policies and managerial guidelines*. World Health Organization.

Organization, W. H., & for International Tobacco Control, R. (2008). *WHO report on the global tobacco epidemic, 2008: the MPOWER package*. World Health Organization.

Organization, W. H., et al. (2014). *Global status report on noncommunicable diseases 2014*. WHO/NMH/NVI/15.1. World Health Organization.

Organization, W. H., et al. (2018). Assessing national capacity for the prevention and control of noncommunicable diseases: report of the 2017 global survey. *World Health Organization (WHO)*.

Organization, W. H., et al. (2020). Assessing national capacity for the prevention and control of noncommunicable diseases: report of the 2019 global survey. *World Health Organization (WHO)*.

Paiva, A. (2010). Hypothesis testing.

Pan, F., Chen, T., Sun, X., Li, K., Jiang, X., Försti, A., Zhu, Y., & Lai, M. (2019). Prognosis prediction of colorectal cancer using gene expression profiles. *Frontiers in oncology*, *9*, 252.

Pan, Q. (2013). Multiple hypotheses testing procedures in clinical trials and genomic studies. *Frontiers in public health*, *1*, 63.

Pappu, V., & Pardalos, P. M. (2014). High-dimensional data classification. In F. Aleskerov, B. Goldengorin, & P. M. Pardalos (Eds.) *Clusters, Orders, and Trees: Methods and Applications*, Springer Optimization and Its Applications, (pp. 119–150). Springer.
URL `https://ideas.repec.org/h/spr/spochp/978-1-4939-0742-7_8.html`

Pedersen, A. B., Mikkelsen, E. M., Cronin-Fenton, D., Kristensen, N. R., Pham, T. M., Pedersen, L., & Petersen, I. (2017). Missing data and multiple imputation in clinical epidemiological research. *Clinical epidemiology*, *9*, 157.

Pereira, J. M., Basto, M., & da Silva, A. F. (2016). The logistic lasso and ridge regression in predicting corporate failure. *Procedia economics and finance*, *39*, 634–641.

Piao, Y., Piao, M., & Ryu, K. H. (2017). Multiclass cancer classification using a feature subset-based ensemble from microrna expression profiles. *Computers in biology and medicine*, *80*, 39–44.

Popovici, V., Budinska, E., Tejpar, S., Weinrich, S., Estrella, H., Hodgson, G., Van Cutsem, E., Xie, T., Bosman, F. T., Roth, A. D., et al. (2012). Identification of a poor-prognosis braf-mutant-like population of patients with colon cancer. *Journal of clinical oncology*, *30*(12), 1288–1295.

Qi, Y. (2012). Random forest for bioinformatics. In *Ensemble machine learning*, (pp. 307–323). Springer.

R Core Team (2020). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria.
URL `https://www.R-project.org/`

R Core Team (2021). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria.
URL `https://www.R-project.org/`

Rai, M. F., Tycksen, E. D., Sandell, L. J., & Brophy, R. H. (2018). Advantages of rna-seq compared to rna microarrays for transcriptome profiling of anterior cruciate ligament tears. *Journal of orthopaedic research®*, *36*(1), 484–497.

Rajeswari, P., & Reena, G. S. (2011). Human liver cancer classification using microarray gene expression data. *International journal of computer applications*, *34*(6), 25–37.

Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.-H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J. P., et al. (2001). Multiclass cancer

diagnosis using tumor gene expression signatures. *Proceedings of the national academy of sciences*, *98*(26), 15149–15154.

Ramirez, R., Chiu, Y.-C., Hererra, A., Mostavi, M., Ramirez, J., Chen, Y., Huang, Y., & Jin, Y.-F. (2020). Classification of cancer types using graph convolutional neural networks. *Frontiers in physics*, *8*, 203.

Ranstam, J., & Cook, J. (2017). Kaplan–meier curve. *Journal of British surgery*, *104*(4), 442–442.

Rashid, M., Vishwakarma, R. K., Deeb, A. M., Hussein, M. A., & Aziz, M. A. (2019). Molecular classification of colorectal cancer using the gene expression profile of tumor samples. *Experimental biology and medicine*, *244*(12), 1005–1016.

RColorBrewer, S., & Liaw, M. A. (2018). Package 'randomforest'. *University of california, berkeley: berkeley, CA, USA*.

Read, D. F., Cook, K., Lu, Y. Y., Le Roch, K. G., & Noble, W. S. (2019). Predicting gene expression in the human malaria parasite plasmodium falciparum using histone modification, nucleosome positioning, and 3d localization features. *PLoS computational biology*, *15*(9), e1007329.

Rezvan, P. H., Lee, K. J., & Simpson, J. A. (2015). The rise of multiple imputation: a review of the reporting and implementation of the method in medical research. *BMC medical research methodology*, *15*(1), 1–14.

Ripley, B., Venables, W., & Ripley, M. B. (2016). Package 'nnet'. *R package version*, *7*(3-12), 700.

Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2010). edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, *26*(1), 139–140.

Robinson, M. D., & Smyth, G. K. (2007). Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics*, *23*(21), 2881–2887.

Rok, B., & Lara, L. (2013). Smote for high-dimensional class-imbalanced data. *BMC Bioinformatics*, *14*(1064).

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, *63*(3), 581–592.

Saeys, Y., Inza, I., & Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. *bioinformatics*, *23*(19), 2507–2517.

Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, *8*(4), e1249.

Salem, H., Attiya, G., & El-Fishawy, N. (2017). Classification of human cancer diseases by gene expression profiles. *Applied soft computing*, *50*, 124–134.

Sánchez-Maroño, N., Alonso-Betanzos, A., & Tombilla-Sanromán, M. (2007). Filter methods for feature selection–a comparative study. In *International Conference on Intelligent Data Engineering and Automated Learning*, (pp. 178–187). Springer.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85–117.

Shityakov, S., Dandekar, T., & Förster, C. (2015). Gene expression profiles and protein–protein interaction network analysis in aids patients with hiv-associated encephalitis and dementia. *HIV/AIDS (Auckland, NZ)*, *7*, 265.

Shu, J. (2019). Prediction based on random survival forest. *American journal of biomedical science & research*, *6*(2).

Siegel, R. L., Miller, K. D., & Jemal, A. (2019). Cancer statistics, 2019. *CA: a cancer journal for clinicians*, *69*(1), 7–34.

Simon, R., Lam, A., Li, M.-C., Ngan, M., Menenzes, S., & Zhao, Y. (2007). Analysis of gene expression data using brb-array tools. *Cancer informatics*, *3*, 117693510700300022.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sk, S., Jabez, J., & Anu, V. M. (2017). The power of deep learning models: applications. *Networks*, (p. 33).

Smyth, G. K. (2004). Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical applications in genetics and molecular biology*, *3*(1).

Society, A. C. (2015). *Cancer facts & figures 2015*. American Cancer Society.

Society, A. C. (2020). Colorectal cancer facts & figures 2020-2022. *Published online*, (p. 48).

Statnikov, A., Henaff, M., Narendra, V., Konganti, K., Li, Z., Yang, L., Pei, Z., Blaser, M. J., Aliferis, C. F., & Alekseyenko, A. V. (2013). A comprehensive evaluation of multicategory classification methods for microbiomic data. *Microbiome*, *1*(1), 1–12.

Stephens, D., & Diesing, M. (2014). A comparison of supervised classification methods for the prediction of substrate type using multibeam acoustic and legacy grain-size data. *PloS one*, *9*(4), e93950.

Sterne, J. A., White, I. R., Carlin, J. B., Spratt, M., Royston, P., Kenward, M. G., Wood, A. M., & Carpenter, J. R. (2009). Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. *British medical journal (Bmj)*, *338*.

Stintzing, S. (2014). Management of colorectal cancer. f1000prime rep. 2014; 6: 108.

Su, C., Li, D., Li, N., Du, Y., Yang, C., Bai, Y., Lin, C., Li, X., & Zhang, Y. (2018). Studying the mechanism of plagl2 overexpression and its carcinogenic characteristics based on 3'-untranslated region in colorectal cancer. *International journal of oncology*, *52*(5), 1479–1490.

Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, *71*(3), 209–249.

Sutton, C. D. (2005). Classification and regression trees, bagging, and boosting. *Handbook of statistics*, *24*, 303–329.

Tabares-Soto, R., Orozco-Arias, S., Romero-Cano, V., Bucheli, V. S., Rodríguez-Sotelo, J. L., & Jiménez-Varón, C. F. (2020). A comparative study of machine learning and deep learning algorithms to classify cancer types based on microarray gene expression data. *PeerJ computer science*, *6*, e270.

Taminau, J., Lazar, C., Meganck, S., & Nowé, A. (2014). Comparison of merging and meta-analysis as alternative approaches for integrative gene expression analysis. *International scholarly research notices*, *2014*.

Tan, A. C., & Gilbert, D. (2003). Ensemble machine learning on gene expression data for cancer classification. *University of glasgow*.

Tarek, S., Abd Elwahab, R., & Shoman, M. (2017). Gene expression based cancer classification. *Egyptian informatics journal*, *18*(3), 151–159.

Taylor, J. M. (2011). Random survival forests. *Journal of thoracic oncology*, *6*(12), 1974–1975.

Tharwat, A. (2020). Classification assessment methods. *Applied Computing and Informatics*.

Torre, L. A., Bray, F., Siegel, R. L., Ferlay, J., Lortet-Tieulent, J., & Jemal, A. (2015). Global cancer statistics, 2012. *CA: a cancer journal for clinicians*, *65*(2), 87–108.

Torre, L. A., Islami, F., Siegel, R. L., Ward, E. M., & Jemal, A. (2017). Global cancer in women: burden and trends. *Cancer epidemiology and prevention biomarkers*, *26*(4), 444–457.

Tran, T., Rekabdar, B., & Ekenna, C. (2021). Deep learning methods in predicting gene expression levels for the malaria parasite. *Frontiers in Genetics*, (p. 1738).

Trawiński, B., Smętek, M., Telec, Z., & Lasota, T. (2012). Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. *International journal of applied mathematics and computer science*, *22*, 867–881.

van Eeghen, E. E., Bakker, S. D., van Bochove, A., & Loffeld, R. J. (2015). Impact of age and comorbidity on survival in colorectal cancer. *Journal of gastrointestinal oncology*, *6*(6), 605.

Van Hulse, J., Khoshgoftaar, T. M., & Napolitano, A. (2007). Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on machine learning*, (pp. 935–942).

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. (2013). *The nature of statistical learning theory*. Springer science & business media.

Venkat, N. (2018). The curse of dimensionality: Inside out. *Dept. of CSIS, BITS Pilani*.

Vickerstaff, V., Omar, R. Z., & Ambler, G. (2019). Methods to adjust for multiple comparisons in the analysis and sample size calculation of randomised controlled trials with multiple primary outcomes. *BMC medical research methodology*, *19*(1), 1–13.

Wang, B., Kumar, V., Olson, A., & Ware, D. (2019a). Reviving the transcriptome studies: an insight into the emergence of single-molecule transcriptome sequencing. *Frontiers in genetics*, *10*, 384.

Wang, H., & Li, G. (2017). A selective review on random survival forests for high dimensional data. *Quantitative bio-science*, *36*(2), 85.

Wang, H., Ma, C., & Zhou, L. (2009a). A brief review of machine learning and its application. In *2009 international conference on information engineering and computer science*, (pp. 1–4). IEEE.

Wang, H., Zhou, Z., Li, Y., Chen, Z., Lu, P., Wang, W., Liu, W., & Yu, L. (2017). Comparison of machine learning methods for classifying mediastinal lymph node metastasis of non-small cell lung cancer from 18 f-fdg pet/ct images. *EJNMMI research*, *7*(1), 1–11.

Wang, J., Tan, A. C., Tian, T., & Eds (2012). *Next generation microarray bioinformatics: methods and protocols*. Humana Press.

Wang, L., Feng, Z., Wang, X., Wang, X., & Zhang, X. (2010). Degseq: an r package for identifying differentially expressed genes from rna-seq data. *Bioinformatics*, *26*(1), 136–138.

Wang, S., Huang, H., Han, R., Chen, J., Jiang, J., Li, H., Liu, G., & Chen, S. (2019b). Bpap1 directly regulates bpdef to promote male inflorescence formation in betula platyphylla× b. pendula. *Tree physiology*, *39*(6), 1046–1060.

Wang, X. (2016). *Next-generation sequencing data analysis*. CRC Press.

Wang, Z., Gerstein, M., & Snyder, M. (2009b). Rna-seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, *10*(1), 57–63.

Way, M. J., Scargle, J. D., Ali, K. M., & Srivastava, A. N. (2012). *Advances in machine learning and data mining for astronomy*. CRC Press Boca Raton, FL.

WCRF, W. C. R. F. (2019). Colorectal cancer statistics. `https://www.wcrf.org/dietandcancer/colorectal-cancer-statistics/`. Accessed: 2019-08-02.

Wei, R., Wang, J., Jia, W., & Wei, M. R. (2018). Package 'multiroc'.

Weigelt, B., Baehner, F. L., & Reis-Filho, J. S. (2010). The contribution of gene expression profiling to breast cancer classification, prognostication and prediction: a retrospective of the last decade. *The journal of pathology: A journal of the pathological society of great Britain and Ireland*, *220*(2), 263–280.

White, A., Ironmonger, L., Steele, R. J., Ormiston-Smith, N., Crawford, C., & Seims, A. (2018). A review of sex-related differences in colorectal cancer incidence,

screening uptake, routes to diagnosis, cancer stage and survival in the uk. *BMC cancer*, *18*(1), 1–11.

White, I. R., Royston, P., & Wood, A. M. (2011). Multiple imputation using chained equations: issues and guidance for practice. *Statistics in medicine*, *30*(4), 377–399.

WHO (2019). Cancer. `https://www.who.int/newsroom/fact-sheets/detail/cancer`. Accessed: 2019-08-14.

WHO (2021a). Cancer. `https://www.who.int/news-room/fact-sheets/detail/cancer`. Accessed: 2021-06-04.

WHO, W. H. O. (2021b). Cancer. `https://www.who.int/news-room/fact-sheets/detail/cancer`. Accessed: 2021-05-25.

Witten, D. M. (2011). Classification and clustering of sequencing data using a poisson model. *The annals of applied statistics*, *5*(4), 2493–2518.

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, *5*(2), 241–259.

Wong, L.-J. C. (2017). *Next generation sequencing based clinical molecular diagnosis of human genetic disorders*. Springer.

Xiao, J., Tang, X., Li, Y., Fang, Z., Ma, D., He, Y., & Li, M. (2011). Identification of microrna precursors based on random forest with network-level representation method of stem-loop structure. *BMC bioinformatics*, *12*(1), 1–8.

Xu, B., Huang, J. Z., Williams, G., Wang, Q., & Ye, Y. (2012). Classifying very high-dimensional data with random forests built from small subspaces. *International journal of data warehousing and mining (IJDWM)*, *8*(2), 44–63.

Yan, Z., Li, J., Xiong, Y., Xu, W., & Zheng, G. (2012). Identification of candidate colon cancer biomarkers by applying a random forest approach on microarray data. *Oncology reports*, *28*(3), 1036–1042.

Yang, P., Hwa Yang, Y., B Zhou, B., & Y Zomaya, A. (2010). A review of ensemble methods in bioinformatics. *Current bioinformatics*, *5*(4), 296–308.

Yang, S., & Naiman, D. Q. (2014). Multiclass cancer classification based on gene expression comparison. *Statistical applications in genetics and molecular biology*, *13*(4), 477–496.

Yang, Z., Yu, Y., You, C., Steinhardt, J., & Ma, Y. (2020). Rethinking bias-variance trade-off for generalization of neural networks. In *International conference on machine learning*, (pp. 10767–10777). PMLR.

Yao, Z., & Ruzzo, W. L. (2006). A regression-based k nearest neighbor algorithm for gene function prediction from heterogeneous data. In *BMC bioinformatics*, vol. 7, (pp. 1–11). BioMed Central.

Zang, C., Wang, T., Deng, K., Li, B., Qin, Q., Xiao, T., Zhang, S., Meyer, C. A., He, H. H., Brown, M., et al. (2016). High-dimensional genomic data bias correction and data integration using mancie. *Nature communications*, *7*(1), 1–8.

Zararsız, G., Goksuluk, D., Korkmaz, S., Eldem, V., Zararsiz, G. E., Duru, I. P., & Ozturk, A. (2017). A comprehensive simulation study on classification of rna-seq data. *PloS one*, *12*(8), e0182507.

Zhang, T.-M., Huang, T., & Wang, R.-F. (2018). Cross talk of chromosome instability, cpg island methylator phenotype and mismatch repair in colorectal cancer. *Oncology letters*, *16*(2), 1736–1746.

Zhang, W., Yu, Y., Hertwig, F., Thierry-Mieg, J., Zhang, W., Thierry-Mieg, D., Wang, J., Furlanello, C., Devanarayan, V., Cheng, J., et al. (2015). Comparison of rna-seq and microarray-based models for clinical endpoint prediction. *Genome biology*, *16*(1), 133.

Zhang, X.-D. (2020). Machine learning. In *A Matrix Algebra Approach to Artificial Intelligence*, (pp. 223–440). Springer.

Zhao, D., Wang, X., Mu, Y., & Wang, L. (2021). Experimental study and comparison of imbalance ensemble classifiers with dynamic selection strategy. *Entropy*, *23*(7), 822.

Zhao, S., Zhang, B., Zhang, Y., Gordon, W., Du, S., Paradis, T., Vincent, M., & von Schack, D. (2016). Bioinformatics for rna-seq data analysis. *Bioinformatics-updated features and applications: InTech*, (pp. 125–49).

Zheng, H., Ying, H., Wiedemeyer, R., Yan, H., Quayle, S. N., Ivanova, E. V., Paik, J.-H., Zhang, H., Xiao, Y., Perry, S. R., et al. (2010). Plagl2 regulates wnt signaling to impede differentiation in neural stem cells and gliomas. *Cancer cell*, *17*(5), 497–509.

Zhou, Z.-H. (2019). *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC.

Ziegler, A., Konig, I. R., & Pahlke, F. (2010). *A statistical approach to genetic epidemiology: concepts and applications, with an e-learning platform*. John Wiley & Sons.

Zumwalt, T. J., Shigeyasu, K., Weng, W., Okugawa, Y., Miyoshi, J., & Goel, A. (2016). The atp6v1c2 (a vacuolar-atpase gene) is a novel early prognosticator for colorectal cancer.

Appendices

Appendix A: The significant genes returned using LASSO with 10-folds cross-validation

Table 1: The 173 significant genes that were returned using LASSO with 10-folds cross-validation.

	baseMean	log2Fold Change	lfcSE	stat	pvalue	padj
ACSL4	3370.81	0.86896	0.0476	18.2555	1.87E-74	5.87E-74
ADAM28	528.011	-0.6334	0.07523	-8.4205	3.75E-17	5.89E-17
ADAM8	1711.32	-1.6217	0.06691	-24.236	9.37E-130	4.72E-129
AFF1	6892.33	1.91525	0.03596	53.2632	0	0
ALPK3	806.889	1.03086	0.06688	15.4126	1.35E-53	3.41E-53
ANKS4B	185.576	-4.4648	0.3449	-12.945	2.50E-38	5.34E-38
ANXA6	9806.23	-0.8436	0.04146	-20.347	4.90E-92	1.80E-91
AQP3	5437.67	-4.0455	0.10407	-38.875	0	0
AQP4	906.546	4.21352	0.14895	28.2886	4.77E-176	3.38E-175
ATF1	1520.23	0.63084	0.0224	28.1681	1.44E-174	1.01E-173
ATP2B4	11058.4	-0.1707	0.04502	-3.791	0.00015007	0.00018147
ATP6V0C	11556.3	-0.2019	0.02699	-7.4806	7.40E-14	1.10E-13
BDH2	2407.05	2.14255	0.04332	49.4594	0	0
C11orf41	550.413	-2.9532	0.09899	-29.835	1.39E-195	1.13E-194

C18orf45	1239.63	-2.9454	0.03854	-76.428	0	0
C1orf186	549.158	-1.0929	0.12461	-8.7706	1.78E-18	2.86E-18
C1orf190	113.125	3.35189	0.06494	51.6182	0	0
C2orf89	603.527	2.286	0.08833	25.8802	1.11E-147	6.40E-147
C5orf53	849.576	1.31961	0.03811	34.6256	1.04E-262	1.29E-261
C6orf138	223.422	4.66633	0.09545	48.8861	0	0
C6orf222	123.793	-1.3166	0.22378	-5.8834	4.02E-09	5.43E-09
C9orf167	1440.81	2.15425	0.04608	46.753	0	0
C9orf91	1988.46	-1.8341	0.03453	-53.118	0	0
CCDC109A	1907.7	-0.1384	0.03195	-4.331	1.48E-05	1.84E-05
CCL13	213.252	1.6253	0.12751	12.7464	3.27E-37	6.87E-37
CCNJL	633.306	1.36706	0.06965	19.6283	8.87E-86	3.08E-85
CDH26	72.9319	-0.8001	0.10072	-7.9431	1.97E-15	3.00E-15
CDX1	673.477	-1.5722	0.0964	-16.309	8.51E-60	2.30E-59
CEACAM6	28629.9	-3.6857	0.14666	-25.131	2.27E-139	1.23E-138
CFL2	1388.26	1.19518	0.0398	30.0259	4.51E-198	3.73E-197
CHRAC1	2544.38	0.15682	0.03415	4.59235	4.38E-06	5.52E-06
CHSY1	3327.68	-1.3737	0.0392	-35.046	4.43E-269	5.71E-268
CLDN3	11251.4	1.37722	0.06213	22.1666	7.22E-109	3.09E-108
CPNE8	456.31	0.57156	0.05593	10.219	1.63E-24	2.90E-24
CTSB	111838	3.01424	0.04021	74.9717	0	0
CTSK	7158.98	-2.5965	0.07718	-33.639	4.45E-248	5.05E-247
CTU1	275.536	0.30631	0.04339	7.05937	1.67E-12	2.42E-12
CXCL17	3781.26	-0.7251	0.14759	-4.9128	8.98E-07	1.15E-06
CYP2C18	63.4312	-4.0536	0.26513	-15.289	9.07E-53	2.27E-52
DCBLD2	4066.26	2.03321	0.05135	39.5971	0	0
DFNA5	605.619	1.6637	0.0525	31.6899	2.14E-220	2.07E-219
DIRAS1	474.766	-2.5309	0.09457	-26.761	9.16E-158	5.69E-157
DOCK3	714.113	4.56055	0.09507	47.9697	0	0

DPY19L1	2490.91	0.73826	0.04509	16.3721	3.03E-60	8.24E-60
DVL3	6057.44	-0.4287	0.0273	-15.702	1.46E-55	3.79E-55
EFS	1994.72	-1.8672	0.0705	-26.485	1.42E-154	8.63E-154
EMX2	1161.03	-1.1471	0.12748	-8.998	2.30E-19	3.75E-19
ENO1	57304.3	-0.3479	0.04262	-8.1623	3.29E-16	5.08E-16
EPHA2	2478.12	2.20884	0.05574	39.6304	0	0
ERO1L	3225.93	-0.3541	0.04669	-7.5841	3.35E-14	5.00E-14
ERRFI1	7027.42	0.95015	0.06015	15.7957	3.33E-56	8.70E-56
F2RL2	946.889	-4.1363	0.09586	-43.152	0	0
FAM115C	640.22	-1.7596	0.05678	-30.989	7.61E-211	6.82E-210
FAM126A	1043.82	-0.6537	0.05516	-11.849	2.17E-32	4.30E-32
FAM49A	457.997	2.93101	0.05577	52.5511	0	0
FOXE1	2538.15	11.0891	0.20785	53.3515	0	0
FUT4	749.106	-0.3538	0.04268	-8.2898	1.13E-16	1.77E-16
FYN	3259.73	2.47148	0.05261	46.9818	0	0
FZD5	1539.99	2.10413	0.05529	38.0591	0	0
GALNT12	1395.26	3.90386	0.07389	52.8336	0	0
GAS2L1	1933.06	-0.8869	0.03449	-25.712	8.53E-146	4.82E-145
GGCX	2233.96	-0.2142	0.03163	-6.77	1.29E-11	1.83E-11
GJD3	218.919	0.76789	0.06063	12.665	9.24E-37	1.93E-36
GMDS	1379.91	-1.0039	0.04837	-20.755	1.11E-95	4.22E-95
GPR35	419.506	-0.7246	0.06243	-11.608	3.77E-31	7.36E-31
GREM1	2587.79	-2.7903	0.10787	-25.868	1.52E-147	8.73E-147
GTF3C3	1987.34	-0.3592	0.02385	-15.062	2.89E-51	7.13E-51
GULP1	640.82	0.34179	0.07153	4.77814	1.77E-06	2.25E-06
HMG20B	5489.95	-0.8186	0.02685	-30.484	4.31E-204	3.71E-203
HMGB3	5577.2	-1.6441	0.06003	-27.388	3.86E-165	2.52E-164
HNF1A	245.61	-1.1339	0.07841	-14.461	2.14E-47	5.07E-47
HNF4G	277.795	-6.2544	0.13349	-46.852	0	0

HOXA13	77.3845	-2.3855	0.20455	-11.662	1.99E-31	3.89E-31
HOXC8	167.777	-5.4351	0.0949	-57.275	0	0
HOXC9	218.878	-5.1496	0.09837	-52.347	0	0
IER5L	2059.74	0.28554	0.05433	5.25538	1.48E-07	1.93E-07
IGF2BP1	142.698	-2.2957	0.18221	-12.6	2.12E-36	4.41E-36
IL22RA1	266.713	2.25289	0.09592	23.4876	5.46E-122	2.56E-121
IPPK	1053.05	-1.0976	0.03032	-36.207	4.81E-287	6.90E-286
IQCA1	828.264	3.48015	0.08491	40.9859	0	0
IRAK3	760.228	-0.8446	0.07413	-11.394	4.46E-30	8.57E-30
IRX5	1464.78	-7.411	0.08945	-82.853	0	0
ISCU	5100.68	1.90679	0.02875	66.3119	0	0
ITGB3	1998.06	4.17437	0.08147	51.2376	0	0
KCNJ15	2194.64	6.55889	0.08335	78.6886	0	0
KLF3	4155.05	0.39248	0.02462	15.9413	3.27E-57	8.63E-57
KLHL14	1062.48	5.21229	0.10642	48.9801	0	0
LILRB3	177.152	-0.478	0.06915	-6.9133	4.74E-12	6.78E-12
LMO7	6171.05	2.30817	0.04991	46.2485	0	0
LOC84740	930.367	-4.7457	0.15509	-30.6	1.21E-205	1.05E-204
LPCAT1	8849.05	-1.1973	0.04956	-24.157	6.28E-129	3.14E-128
MAFK	2562.79	0.24663	0.03922	6.28828	3.21E-10	4.44E-10
MAPKAPK3	3705.76	1.74455	0.03236	53.9119	0	0
MEIS1	2136.62	-1.7935	0.05538	-32.387	4.14E-230	4.25E-229
MUC4	1639.02	-0.5678	0.15128	-3.7534	0.000174446	0.000210365
NBPF10	2075.07	-1.1899	0.05655	-21.041	2.74E-98	1.07E-97
NIN	2548.78	-0.8453	0.03187	-26.528	4.66E-155	2.83E-154
NUDT16P1	458.128	1.45315	0.06586	22.0633	7.11E-108	3.02E-107
NYNRIN	2599.49	1.2127	0.05817	20.8465	1.64E-96	6.29E-96
OGFRL1	1005.35	-1.5405	0.05503	-27.992	2.02E-172	1.39E-171
OSBPL3	1900.07	1.40025	0.0519	26.9776	2.70E-160	1.71E-159

PABPC3	1407.18	-0.9747	0.04857	-20.067	1.43E-89	5.15E-89
PADI2	2795.44	-4.4786	0.09602	-46.642	0	0
PAPLN	1439.99	2.47032	0.0647	38.1831	0	0
PCDP1	155.199	1.20581	0.13261	9.093	9.63E-20	1.58E-19
PIK3R1	5504.79	-0.687	0.05038	-13.637	2.43E-42	5.44E-42
PITPNM1	3075.04	1.17408	0.04443	26.4284	6.46E-154	3.89E-153
PNRC1	6180.57	0.64155	0.03687	17.4	8.25E-68	2.43E-67
POF1B	1116.43	-3.0652	0.12466	-24.588	1.70E-133	8.85E-133
POSTN	36945.3	-3.1331	0.09535	-32.86	8.24E-237	8.75E-236
PPP2R4	10291.5	-0.539	0.02472	-21.806	2.03E-105	8.42E-105
PPP4R1	4565.9	0.19383	0.0274	7.07291	1.52E-12	2.19E-12
PRIMA1	426.151	-1.4497	0.10554	-13.737	6.11E-43	1.38E-42
PROX1	94.9841	1.67423	0.10665	15.6988	1.54E-55	3.99E-55
PRR7	356.667	-0.7432	0.06295	-11.807	3.59E-32	7.08E-32
PRSS3	347.489	1.20214	0.168	7.15547	8.34E-13	1.21E-12
PSME4	5848.39	-0.896	0.03328	-26.926	1.08E-159	6.81E-159
PTGS1	3620.61	-1.3029	0.05834	-22.333	1.77E-110	7.69E-110
RAB11FIP5	2452.82	-0.6883	0.03307	-20.812	3.35E-96	1.28E-95
RAB3IP	1843.85	-0.4525	0.04814	-9.399	5.51E-21	9.25E-21
RBM14	3005.43	-0.1571	0.01623	-9.6794	3.69E-22	6.32E-22
RBMXL1	1757.42	-0.1196	0.02259	-5.2947	1.19E-07	1.56E-07
REP15	67.4638	0.57378	0.06699	8.56567	1.07E-17	1.71E-17
RHOF	1401.89	1.75816	0.07418	23.7008	3.54E-124	1.69E-123
RHOU	3104.79	2.1605	0.06005	35.9766	1.94E-283	2.71E-282
ROS1	587.327	-1.0717	0.22531	-4.7565	1.97E-06	2.51E-06
RTN4RL1	820.116	-4.1761	0.09025	-46.273	0	0
SEL1L3	5779.97	2.04867	0.06482	31.6048	3.17E-219	3.04E-218
SFTA2	860.678	4.31509	0.18356	23.5073	3.43E-122	1.61E-121
SFTPA2	26597.7	4.01867	0.15964	25.1727	7.98E-140	4.34E-139

SFTPB	58184.4	10.7305	0.18489	58.0387	0	0
SGPP2	505.305	1.69707	0.0884	19.198	3.85E-82	1.29E-81
SH3TC2	137.14	-0.3047	0.08531	-3.5711	0.000355476	0.000424991
SHOX2	137.388	-2.6585	0.08671	-30.659	2.01E-206	1.75E-205
SLC16A3	4475.32	0.2324	0.06165	3.76938	0.000163654	0.000197543
SLC5A6	3718.44	-3.0248	0.0499	-60.621	0	0
SNRPN	7199.12	2.20516	0.05457	40.4128	0	0
SNTB1	2940.19	2.87747	0.0581	49.5222	0	0
SOX17	1447.5	0.32306	0.05814	5.55664	2.75E-08	3.64E-08
SOX2	328.115	-5.2739	0.18675	-28.241	1.86E-175	1.31E-174
SPRR3	50.4587	-3.0802	0.2934	-10.498	8.83E-26	1.59E-25
SRL	342.905	5.44947	0.07675	71.0073	0	0
STAMBPL1	350.138	-0.6987	0.05128	-13.624	2.87E-42	6.43E-42
STARD3NL	1970.17	0.60911	0.02799	21.7602	5.53E-105	2.29E-104
STK17A	1733.47	-0.4008	0.041	-9.7763	1.42E-22	2.46E-22
STK33	306.893	4.99973	0.10542	47.4282	0	0
TBX4	73.5169	-3.1685	0.15579	-20.339	5.82E-92	2.13E-91
TBX5	141.449	-2.4714	0.10744	-23.003	4.34E-117	1.97E-116
TFAP2A	3570.45	-7.265	0.07658	-94.868	0	0
TFPI	2180.11	-0.8757	0.07368	-11.886	1.41E-32	2.79E-32
TG	122733	14.6101	0.09508	153.656	0	0
TMEM125	1976.24	0.60393	0.0452	13.3617	1.01E-40	2.23E-40
TMEM189.UBE2V1	106.474	-0.9133	0.18485	-4.9407	7.78E-07	1.00E-06
TMPRSS4	3172.42	3.24068	0.13923	23.2751	7.92E-120	3.68E-119
TMUB1	2457.84	0.38033	0.02643	14.3878	6.17E-47	1.45E-46
TNFSF10	8460.82	-2.8268	0.07024	-40.246	0	0
TP53I3	1034.62	0.86222	0.04468	19.2996	5.41E-83	1.83E-82
TRPS1	14055.4	-4.6335	0.05918	-78.295	0	0
TSHR	5366.57	10.606	0.10119	104.809	0	0

TSHZ2	342.181	-1.2576	0.07492	-16.785	3.12E-63	8.76E-63
TSPAN3	15911.8	1.43967	0.03965	36.3094	1.15E-288	1.67E-287
TXNRD1	6856.62	-0.6134	0.04865	-12.61	1.87E-36	3.89E-36
UAP1	4251.85	-1.5417	0.03362	-45.86	0	0
UBAP1	3893.18	-0.0895	0.02371	-3.7774	0.000158448	0.000191353
VANGL1	2792.57	-1.1929	0.03663	-32.563	1.37E-232	1.43E-231
VSTM2L	1170.07	1.22453	0.11733	10.4363	1.69E-25	3.04E-25
WT1	1797.4	-4.7802	0.1424	-33.569	4.83E-247	5.48E-246
XAGE1D	1129.49	-4.7626	0.30729	-15.499	3.53E-54	9.00E-54
ZDHHC7	3272.22	-0.1507	0.02559	-5.8885	3.90E-09	5.27E-09
ZNF280B	198.397	2.01472	0.0844	23.8724	5.93E-126	2.89E-125
ZNF628	562.854	0.27863	0.0287	9.70931	2.75E-22	4.73E-22
ZNF771	256.77	-0.2152	0.04531	-4.7499	2.04E-06	2.59E-06
ZNF90	1110.07	1.3678	0.06389	21.4085	1.11E-101	4.47E-101

Appendix B: Oversampling Results

The overall predictive performance of the machine learning methods based on the oversampling

Table 2: The overall predictive performance of the machine learning methods based on the oversampling.

Methods	Performance Measures					
	ACC (95% CI)	**Kappa (95% CI)**	**F1-Score**	**Precision**	**Sensitivity**	**AUC**
SVM-R	93.1 (90.8, 94.9)	89.6 (86.8, 92.5)	97.6	99.4	95.9	97.2
SVM-L	82.4 (79.3, 85.3)	71.9 (67.5, 76.3)	88.1	100.0	78.7	92.0
SVM-P	84.0 (80.9, 86.7)	75.8 (71.9, 79.6)	94.7	100.0	89.9	90.9
ANN	86.3 (83.4, 88.8)	80.4 (76.8, 84.1)	92.1	86.9	97.9	89.7
kNN	92.0 (89.6, 94.0)	88.4 (85.4, 91.3)	96.0	93.6	98.4	96.3
Bagging	98.0 (96.6, 98.9)	97.0 (95.4, 98.6)	98.1	100	96.2	99.4

Note: SVM-R, Support Vector Machine with Radial-basis function (RBF) kernel; SVM-L, Support Vector Machine with Linear Kernel; SVM-P, Support Vector Machine with Polynomial Kernel; ANN, Artificial Neural Networks; kNN, K-nearest Neighbors; ACC, Accuracy; CI, Confidence Interval; Kappa, Kappa Statistics; AUC, Area Under the Curve.

Predictive performance of the machine learning methods per cancer type based on the oversampling

Table 3: Predictive performance of the machine learning methods per-class statistics based on the oversampling.

Performance		Methods					
Measures	Class	SVM-R	SVM-L	SVM-P	ANN	kNN	Bagging trees
	BRCA	97.5	86.1	94.2	92.5	96.0	98.0
	COAD	93.1	97.5	94.9	90.9	93.7	95.8
Accuracy	LUAD	97.9	82.2	89.8	80.1	86.0	97.1
	OV	82.7	52.3	50.6	93.2	98.0	97.1
	THCA	96.0	99.6	99.6	98.9	99.1	100.0
	BRCA	99.4	100.0	100.0	86.9	93.6	100.0
	COAD	86.1	97.2	100.0	88.9	94.4	91.7
Sensitivity	LUAD	100.0	66.3	80.2	61.6	72.1	94.2
	OV	66.7	4.6	01.1	90.8	96.6	94.3
	THCA	91.9	99.1	99.1	99.1	98.2	100.0
	BRCA	95.6	72.2	88.4	98.1	98.4	95.9
	COAD	100.0	97.7	89.7	93.0	93.0	100.0
Specificity	LUAD	95.7	98.0	99.3	98.6	99.8	100.0
	OV	98.8	100.8	100.0	95.6	99.5	100.0
	THCA	100.0	100.0	100.0	98.7	100.0	100.0
	BRCA	97.6	88.1	94.7	92.1	96.0	98.1
	COAD	92.5	82.4	53.3	57.7	60.2	95.7
F1-Score	LUAD	87.8	74.0	86.8	72.1	83.2	97.0
	OV	76.3	08.8	02.3	82.7	96.6	97.0
	THCA	95.8	99.6	99.6	96.5	99.1	100.0
	BRCA	95.9	78.7	89.9	98.0	98.4	96.2
	COAD	100.0	71.4	36.4	42.7	44.2	100.0
Precision	LUAD	78.2	83.8	94.5	86.9	98.4	100.0
	OV	89.2	100.1	100.0	76.0	96.6	100.0
	THCA	100.0	100.0	100.0	94.0	100.0	100.0

Note: SVM-R, Support Vector Machine with Radial-basis function (RBF) kernel; SVM-L, Support Vector Machine with Linear Kernel; SVM-P, Support Vector Machine with Polynomial Kernel; ANN, Artificial Neural Networks; kNN, K-nearest Neighbors.

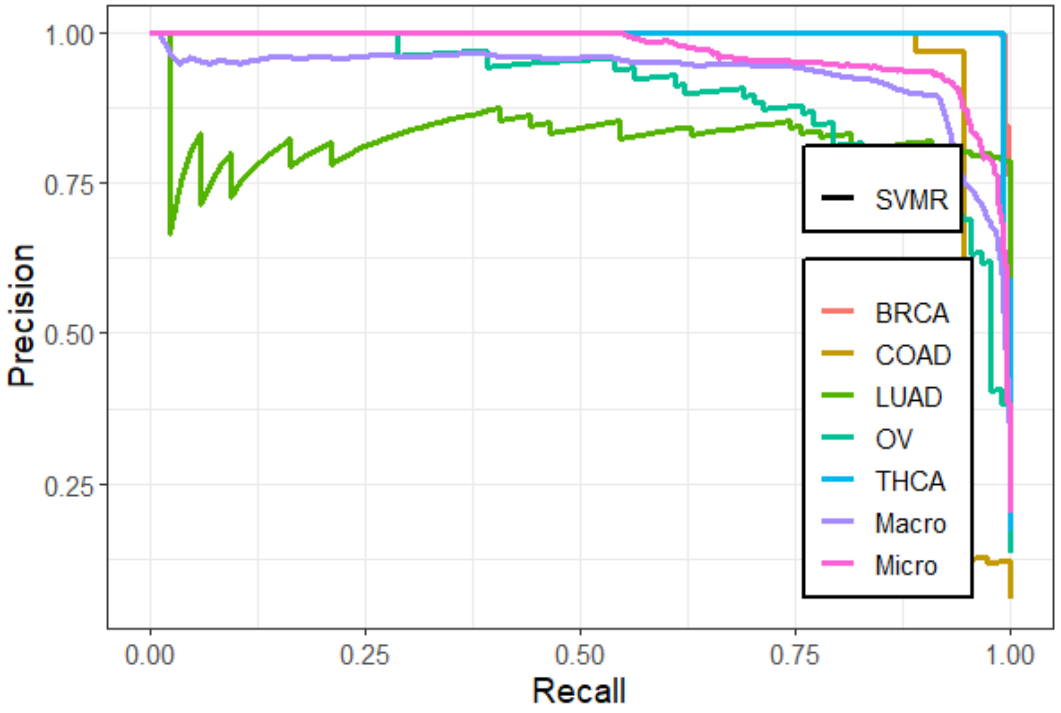


Figure 1: Multi-class Precision Recall Curves visualization for SVMR model based on oversampling technique.

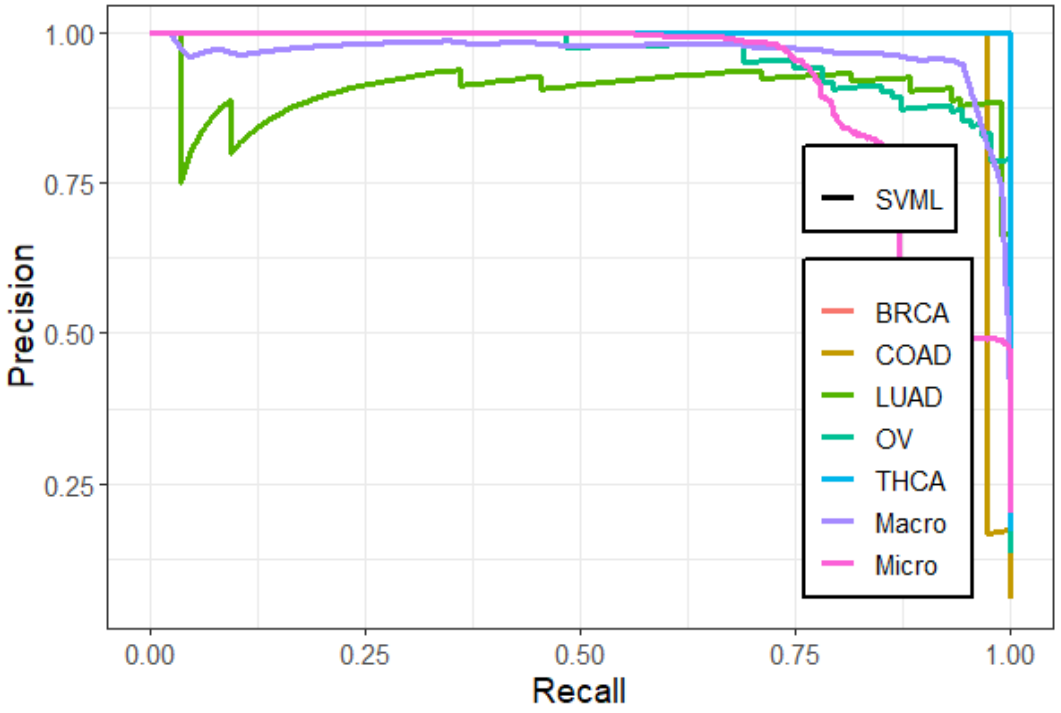


Figure 2: Multi-class Precision Recall Curves visualization for SVMRL model based on oversampling technique.

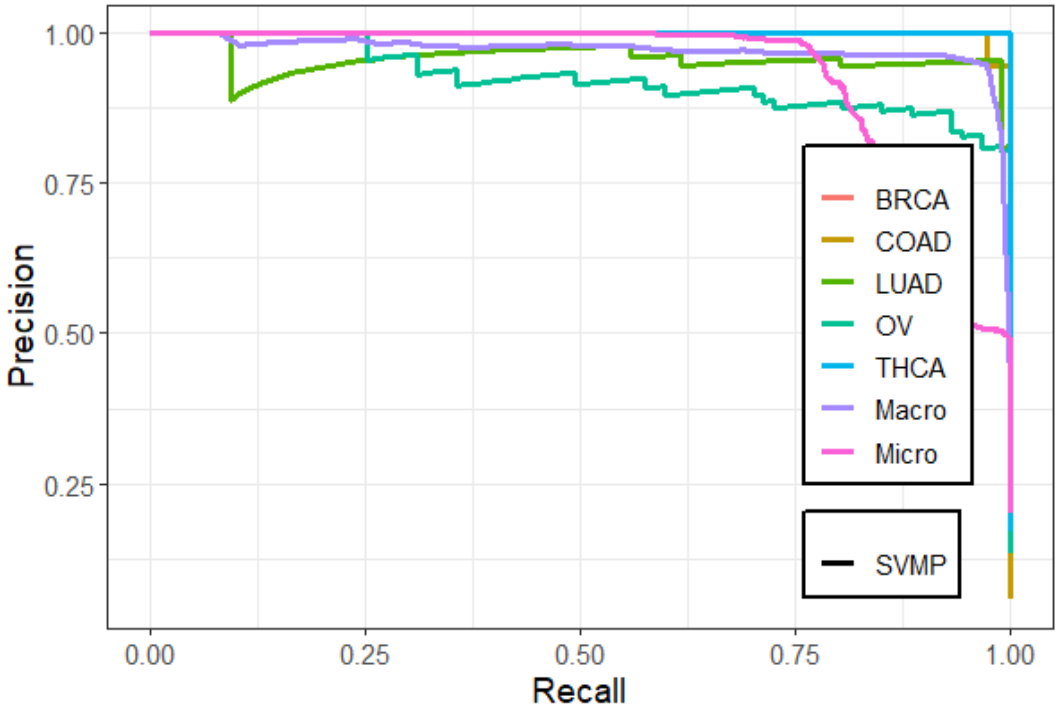


Figure 3: Multi-class Precision Recall Curves visualization for SVMP model based on oversampling technique.

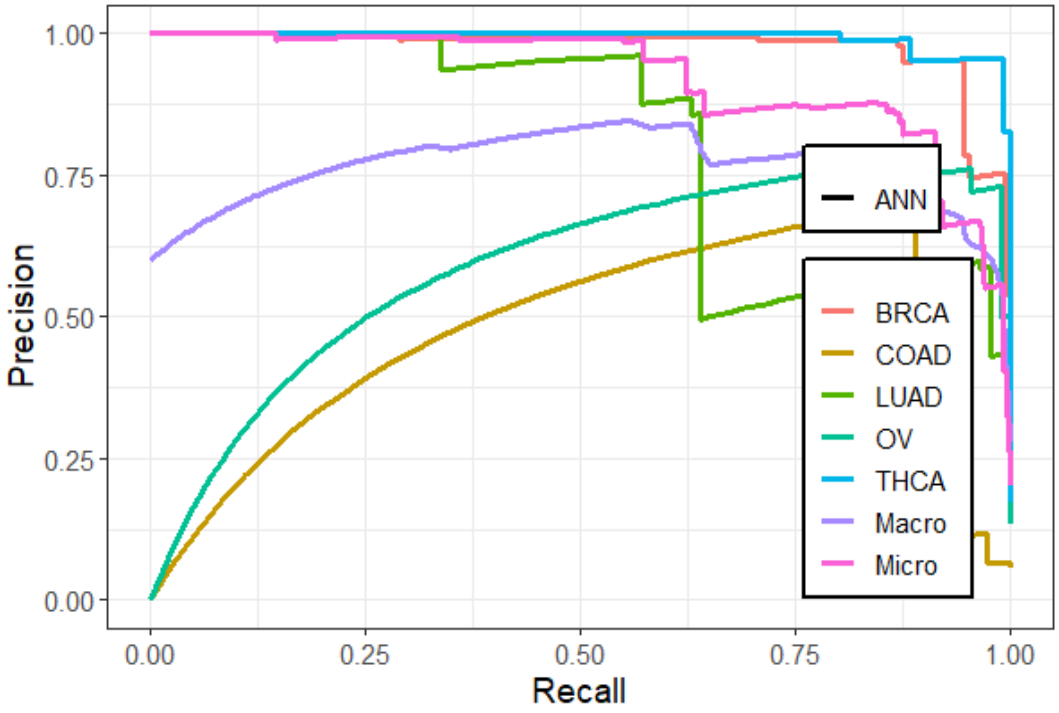


Figure 4: Multi-class Precision Recall Curves visualization for ANN model based on oversampling technique.

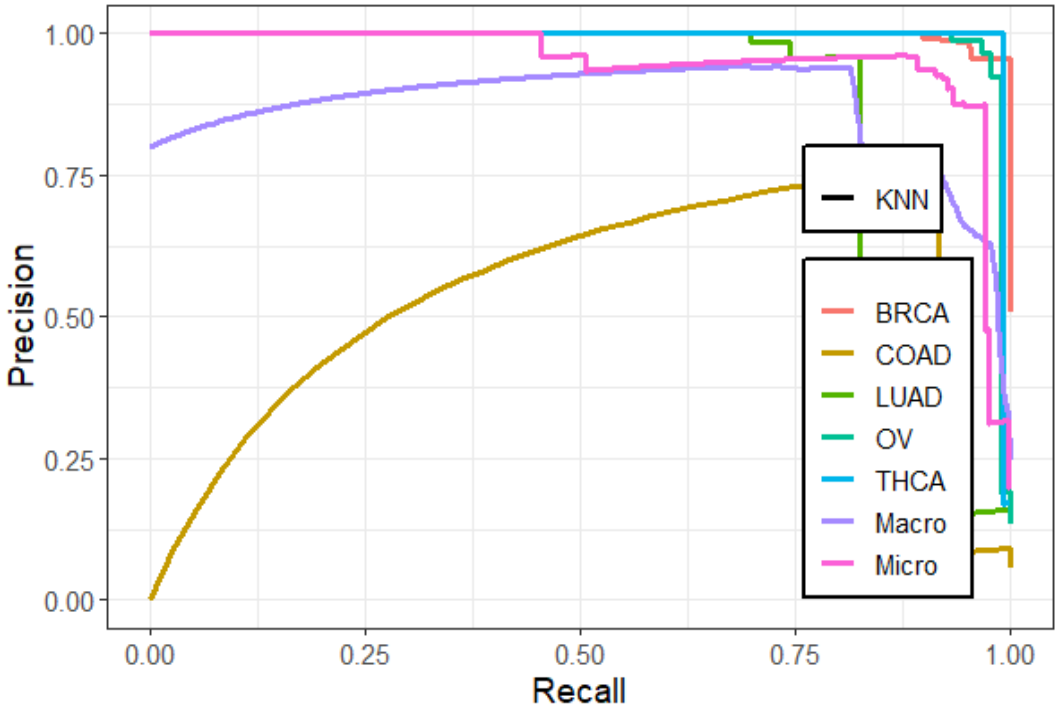


Figure 5: Multi-class Precision Recall Curves visualization for KNN model based on over-sampling technique.

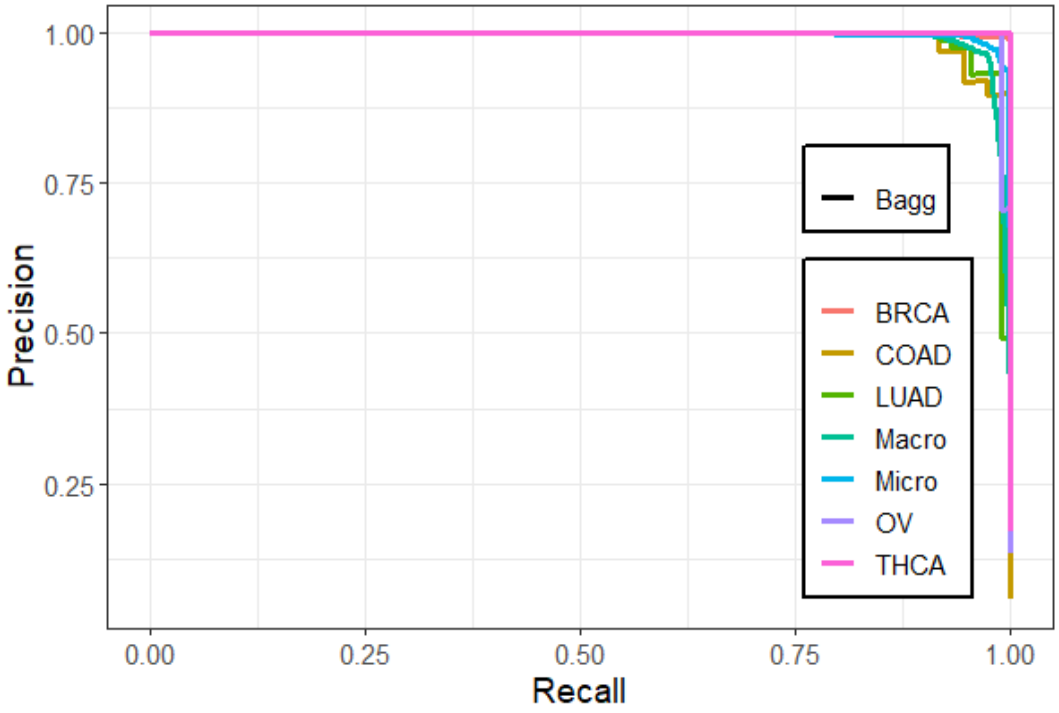


Figure 6: Multi-class Precision Recall Curves visualization for bagging trees model based on over-sampling technique.

Statistical Significance Test

Table 4: Predictive performance of the machine learning methods per-class statistics based on the oversampling.

Accuracy						
	SVMR	**SVML**	**SVMP**	**ANN**	**KNN**	**Bagging**
SVMR		0.120	0.109	0.090	-0.008	-0.054
SVML	0.001		-0.011	-0.030	-0.128	-0.174
SVMP	0.001	1.00		-0.019	-0.117	-0.163
ANN	0.002	1.00	1.00		-0.098	-0.144
KNN	1.00	<0.001	<0.001	0.004		-0.045
Bagging	0.077	<0.001	<0.001	<0.001	<0.001	
Kappa						
	SVMR	**SVML**	**SVMP**	**ANN**	**KNN**	**Bagging**
SVMR		0.193	0.162	0.135	-0.013	-0.079
SVML	<0.001		-0.030	-0.057	-0.206	-0.272
SVMP	<0.001	1.00		-0.027	-0.175	-0.241
ANN	0.002	1.00	1.00		-0.148	-0.214
KNN	1.00	<0.001	<0.001	0.004		-0.066
Bagging	0.076	<0.001	<0.001	<0.001	<0.001	

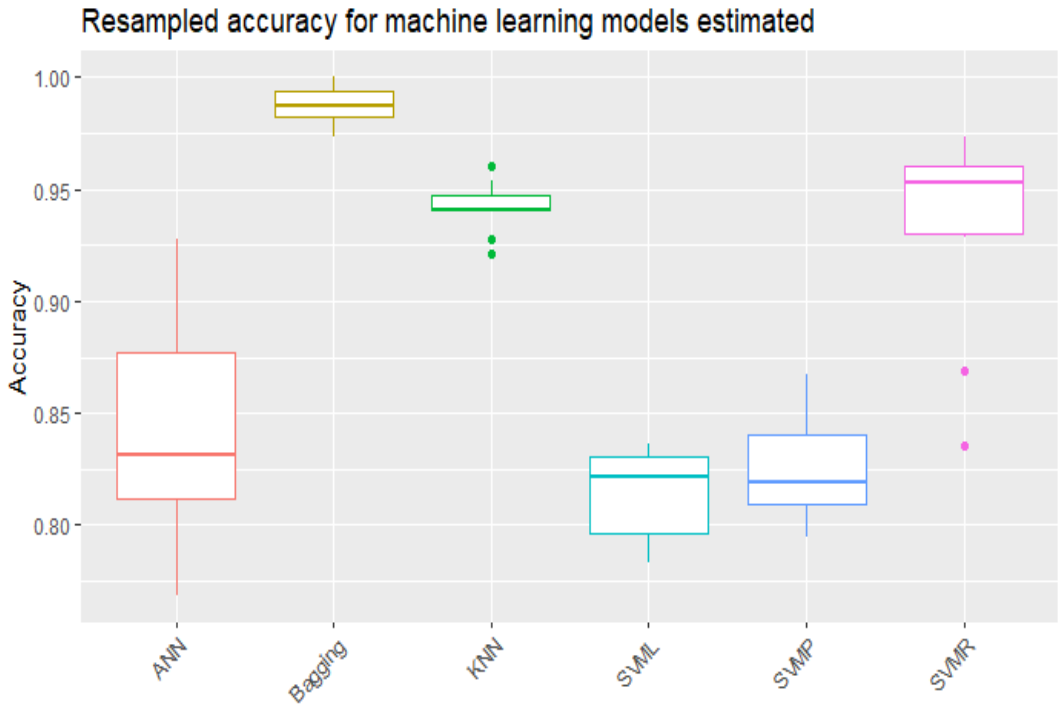


Figure 7: Compares both the mean estimated accuracy and kappa statistic as well as the 95% confidence interval for the methods based on the over-sampling technique.

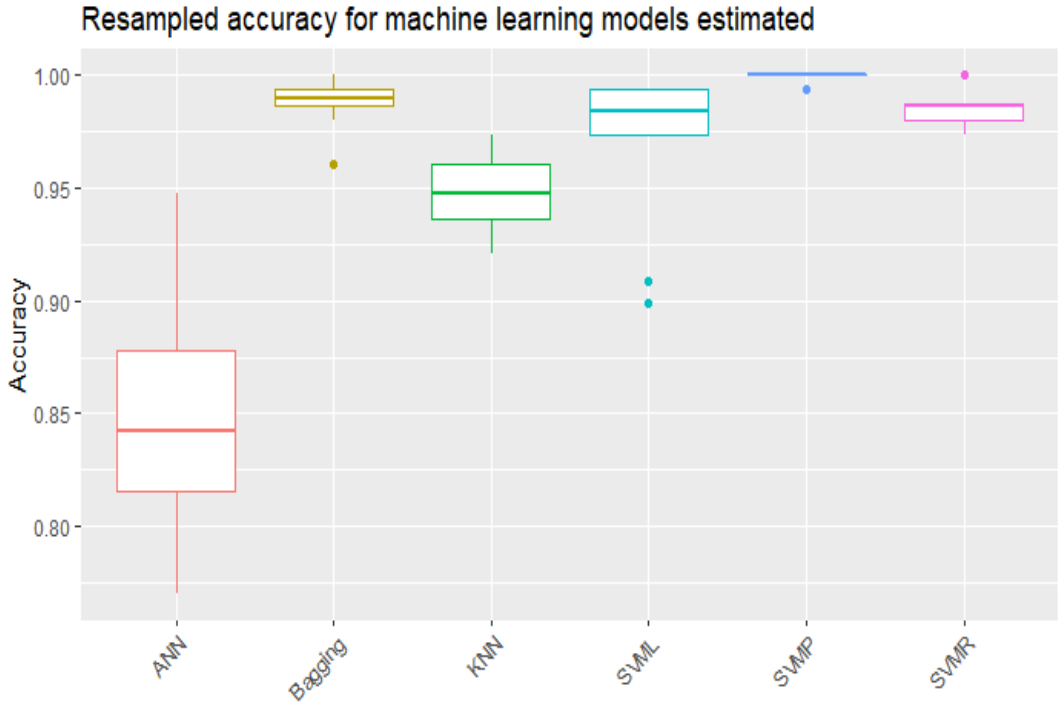


Figure 8: Compares both the mean estimated accuracy and kappa statistic as well as the 95% confidence interval for the methods based on the under-sampling technique.

Appendix C: Summary statistics of the 54 genes

Table 5: Summary statistics of the 54 genes selected for survival analysis ($N = 307$).

Probeset ID*	Gene Symbol	Min	Max	Mean (SD)
200661_at	CTSA	3.085296	3.537954	3.32 (0.09)
202949_s_at	FHL2	3.043399	3.51561	3.33 (0.08)
203725_at	GADD45A	2.841759	3.425452	3.10 (0.10)
204014_at	DUSP4	1.611363	3.343607	2.49 (0.36)
204073_s_at	MYRF	1.931487	3.118863	2.63 (0.25)
204653_at	TFAP2A	1.651537	3.271069	2.30 (0.34)
205767_at	EREG	1.330575	3.480803	2.71 (0.51)
206034_at	SERPINB8	2.2372	2.910791	2.57 (0.12)
206574_s_at	PTP4A3	2.234643	3.305859	2.77 (0.22)
206907_at	TNFSF9	1.82003	3.094666	2.34 (0.24)
207033_at	GIF	1.443817	3.342563	1.91 (0.30)
207267_s_at	RIPPLY3	1.415345	2.595129	1.84 (0.20)
207519_at	SLC6A4	1.860941	2.825926	2.20 (0.19)
209016_s_at	KRT7	1.894892	3.590677	2.42 (0.25)
210074_at	CTSV	2.406115	3.301904	2.87 (0.18)
210306_at	L3MBTL1	1.533169	2.845164	2.12 (0.25)
212947_at	SLC9A8	1.970226	2.87913	2.53 (0.13)
213499_at	CLCN2	2.205234	3.07652	2.58 (0.13)
218056_at	BFAR	2.900836	3.240013	3.08 (0.06)
218611_at	IER5	2.846535	3.438737	3.10 (0.11)
218641_at	C11orf95	2.273865	3.221723	2.87 (0.16)
219232_s_at	EGLN3	1.995892	3.25081	2.63 (0.23)
219281_at	MSRA	2.463159	3.096048	2.85 (0.10)
219459_at	POLR3B	2.164917	3.010461	2.75 (0.11)
219973_at	ARSJ	1.477639	2.882906	2.17 (0.25)

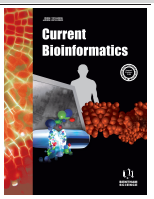
220363_s_at	ELMO2	2.207412	2.797861	2.52 (0.11)
220606_s_at	ADPRM	2.198647	2.884304	2.53 (0.13)
220668_s_at	DNMT3B	2.078965	2.929752	2.52 (0.15)
220736_at	SLC19A3	1.541837	3.07391	2.13 (0.32)
221522_at	ANKRD27	2.839777	3.303317	3.08 (0.08)
221605_s_at	PIPOX	1.599295	3.263056	2.21 (0.32)
224368_s_at	NDRG3	2.274761	3.136462	2.80 (0.16)
224916_at	TMEM173	2.209758	3.181648	2.79 (0.15)
225923_at	VAPB	2.391236	3.181648	2.80 (0.15)
226004_at	CABLES2	2.359722	3.15781	2.80 (0.13)
227134_at	SYTL1	1.970423	3.184009	2.67 (0.22)
227949_at	PHACTR3	1.159216	2.99362	1.86 (0.37)
228262_at	MAP7D2	1.340846	3.273704	2.27 (0.50)
229522_at	SDR42E1	2.066187	2.872696	2.53 (0.16)
230084_at	SLC30A2	1.996567	2.894033	2.32 (0.16)
232277_at	SLC28A3	1.367166	2.942343	2.07 (0.39)
232652_x_at	SCAND1	2.865234	3.359162	3.09 (0.10)
232884_s_at	ZNF853	1.813342	2.785865	2.15 (0.18)
233979_s_at	ESPN	1.901582	2.90773	2.39 (0.20)
234725_s_at	SEMA4B	2.722612	3.327446	3.08 (0.11)
234728_s_at	DHX35	1.992349	2.876806	2.47 (0.16)
234985_at	LDLRAD3	1.974432	3.167782	2.74 (0.21)
235515_at	SYNE4	1.868786	3.038119	2.57 (0.28)
235798_at	TMEM170B	1.180868	2.532567	1.82 (0.27)
236514_at	ACOT8	1.741381	3.000081	2.36 (0.25)
238824_at	RPS29	1.95044	2.952403	2.44 (0.14)
238935_at	RPS27L	1.804659	3.16445	2.60 (0.19)
242963_at	SGMS2	1.550172	2.834302	2.23 (0.22)
32502_at	GDPD5	2.424871	3.357056	2.86 (0.15)

Appendix D: Published Papers

Current Bioinformatics, 2021, *16*, 583-600

RESEARCH ARTICLE

Colorectal Cancer Classification and Survival Analysis Based on an Integrated RNA and DNA Molecular Signature



Mohanad Mohammed[1,*], Henry Mwambi[1] and Bernard Omolo[1,2,3]

[1]*School of Mathematics, Statistics, and Computer Science, University of KwaZulu-Natal, Pietermaritzburg, Private Bag X01, Scottsville 3209, South Africa;* [2]*Division of Mathematics & Computer Science, University of South Carolina-Upstate, 800 University Way, Spartanburg, USA,* [3]*School of Public Health, Faculty of Health Sciences, University of the Witwatersrand, Johannesburg, South Africa*

ARTICLE HISTORY

Received: April 19, 2020
Revised: April 24, 2020
Accepted: June 06, 2020

DOI:
10.2174/1574893615999200711170445

Abstract: ***Background*:** Colorectal cancer (CRC) is the third most common cancer among women and men in the USA, and recent studies have shown an increasing incidence in less developed regions, including Sub-Saharan Africa (SSA). We developed a hybrid (DNA mutation and RNA expression) signature and assessed its predictive properties for the mutation status and survival of CRC patients.

***Methods*:** Publicly-available microarray and RNASeq data from 54 matched formalin-fixed paraffin-embedded (FFPE) samples from the Affymetrix GeneChip and RNASeq platforms, were used to obtain differentially expressed genes between mutant and wild-type samples. We applied the support-vector machines, artificial neural networks, random forests, k-nearest neighbor, naïve Bayes, negative binomial linear discriminant analysis, and the Poisson linear discriminant analysis algorithms for classification. Cox proportional hazards model was used for survival analysis.

***Results*:** Compared to the genelist from each of the individual platforms, the hybrid genelist had the highest accuracy, sensitivity, specificity, and AUC for mutation status, across all the classifiers and is prognostic for survival in patients with CRC. NBLDA method was the best performer on the RNASeq data, while the SVM method was the most suitable classifier for CRC across the two data types. Nine genes were found to be predictive of survival.

***Conclusion*:** This signature could be useful in clinical practice, especially for colorectal cancer diagnosis and therapy. Future studies should determine the effectiveness of integration in cancer survival analysis and the application on unbalanced data, where the classes are of different sizes, as well as on data with multiple classes.

Keywords: Colorectal cancer, FFPE, microarray, RAS pathway signatures, RNASeq, molecular.

1. INTRODUCTION

Colorectal cancer (CRC) is one of the major emerging causes of mortality and morbidity around the world [1]. CRC is also the third leading cause of death among men and women [2-6]. According to the World Health Organization (WHO), there were about 1.80 million new cases and 862,000 deaths in the year 2018 [7]. Furthermore, in 2019, CRC was reported to be the third most prevalent cancer among men and women and an estimated 101,420 and 44,180 new cases of colon and rectal cancer, respectively, and 51,020 deaths in the USA alone [2, 8, 9].

Although the incidence rates of CRC are lower in developing countries than in developed countries, recent studies have shown an increase in the incidence rates in Sub-Saharan Africa [10]. Many cancer types that are relatively curable in developed countries are detected only at advanced stages in developing countries, due to late or inaccurate diagnoses [11]. Cancer tumor classification based on morphological characteristics alone has been shown to have serious limitations in some studies [12]. Physicians aim to diagnose CRC as early as possible to design optimal treatment strategies that are patient-specific. Therefore, using genetic mutation and features of the tumor would most probably lead to better understanding and early detection of the disease and lead to finding suitable and targeted strategies [13].

Previously, most of the cancer classification research was based on clinical features of the tumors, which lacked the accurate diagnostic ability, hence the need to develop new methods that will better address this critical problem [12, 14]. Recently, DNA microarray technology has greatly

*Address correspondence to this author at the School of Mathematics, Statistics, and Computer Science, University of KwaZulu-Natal, Pietermaritzburg, Private Bag X01, Scottsville 3209, South Africa; E-mail: mohanadadam32@gmail.com

2212-392X/21 $65.00+.00

improved the classification of diseases into sub-types, particularly cancer. This technology allows the processing of thousands of genes simultaneously, hence providing critical information about a disease [15, 16]. Microarray gene expression data have been used widely for cancer detection, prediction, and diagnosis [17]. In the last decade, next-generation sequencing (NGS) technology has emerged as an advancement in cancer and other disease research, based on RNA sequencing methodology. NGS platforms that are most common include Illumina, SOLiD, Ion Torrent semi-conductor sequencing, and single-molecule real-time sequencing [18].

NGS technology has been the most attractive, and its application dramatically improved over the last few years. This technology is high-throughput and has become popular in the detection and analysis of differentially expressed genes [18, 19]. More recently, RNASeq data has been shown to be better than microarray data in terms of quality and accuracy in estimating transcript abundance. However, the two methodologies are different in design and implementation [19-21]. Although RNASeq experiments are expensive, in contrast, they have many advantages over microarrays. RNASeq allows detecting the variation of a single nucleotide, does not require genomic sequence knowledge, provides quantitative expression levels, provides isoform-level expression measurements, and offers a broader dynamic range than microarrays [20]. Moreover, RNASeq allows the detection of novel transcripts, low background signal, and increased specificity and sensitivity [22]. However, our view is that integrated use of data from both technologies may be the best approach, given the available information from both technologies.

Microarray and RNASeq technologies produce gene expression data in different forms. The structure of gene expression produced using microarrays is continuous data, while RNASeq provides a discrete Type of data [23]. What is common between the two technologies is that both generate big datasets consisting of a few sample sizes, where each sample has a large number of genes. Many areas of research, such as clinical, medical, biological, and agriculture, apply the RNASeq technology [24, 25].

Many statistical and machine learning methods have been used to analyze and extract information from massive amounts of gene expression data. These methods include the Poisson linear discriminant analysis (PLDA), negative binomial linear discriminant analysis (NBLDA), support vector machines (SVM), artificial neural networks (ANN), linear discriminant analysis (LDA), and random forests (RF).

These methods have been used and examined in many studies based on RNASeq and microarray data. For example, Aziz *et al.* [26] assessed the ANN performance based on microarray data using six hybrid feature selection methods. Five gene expression datasets were used for evaluating these methods and for understanding how these methods can improve the performance of ANN. Statistical hypothesis tests were used to check the differences between these methods. They showed that the combination of independent component analysis (ICA) and genetic bee colony algorithm had superior performance. Salem *et al.* [27] proposed a new methodology for gene expression data analysis. They combined information gain (IG) and standard genetic algorithm (SGA) for feature selection and reduction, respectively. Their approach was tested on seven cancer datasets and then compared with the most recent approaches. Their results show that the proposed approach outperformed the most recent approaches. Jain *et al.* [28] presented a two-phase hybrid method for cancer classification using eleven microarray datasets for different cancer types. They combined correlation-based feature selection (CFS) and improved-binary particle swarm optimization (IBPSO). Naive Bayes with 10-fold cross-validation was used for assessment. Results indicated that their approach had better performance in terms of accuracy and the number of selected genes.

Anders and Huber [29] conducted differential expression analysis based on the negative binomial distribution, with variance and mean linked by local regression, for count data. Their proposed method controls the type I error and gives good detection power. Zararsiz *et al.* [23] presented a comprehensive simulation study on RNASeq classification using PLDA, NBLDA, single SVM, bagging SVM (bagSVM), classification and regression trees (CART), and RF. Their simulation results were applied and compared to two miRNA and two mRNA real experimental datasets. They found that the power-transformed PLDA, RF, and SVM were the best in classification performance.

Due to the small number of samples for gene expression data, combining independent datasets is novel in order to increase sample size and statistical power. Taminau *et al.* [30] worked on the integration of gene expression analysis using two approaches based on merging and meta-analysis. They used six gene expression datasets. Results showed that both meta-analysis and merging did well, but merging was able to detect more differentially expressed genes than meta-analysis.

Recently, combining two different gene expression data sources has been shown to improve classification accuracy as opposed to using only one source. Castillo and co-workers [20] introduced the integration of multiple microarrays and RNASeq platforms. They first carried out a differential expression analysis, then applied the minimum-redundancy maximum-relevance (mRMR) feature selection approach for further reduction of the gene-list. The top 10 genes were selected and evaluated using four classification methods: k-nearest neighbor (KNN), naive Bayes (NB), RF, and SVM. Their results showed the highest accuracy and *f1*-score for the KNN. In this study, we combined RNASeq and DNA expression data from colorectal cancer patients. We obtained a hybrid gene-list from the RNASeq and microarray datasets and assessed its classification performance based on the PLDA, NBLDA, SVM, RF, ANN, KNN, and NB algorithms.

The paper is structured as follows. Section 2 discusses the methods and the datasets used in the study. Section 3 shows the classification results of the microarray, RNASeq, hybrid gene lists, and survival analysis. Discussion and conclusions are presented in Sections 4 and 5, respectively.

2. MATERIALS AND METHODS

2.1. Datasets

We used publicly available microarray and RNASeq data that is also reported in Omolo *et al.* [4]. The data consists of 54 matched formalin-fixed paraffin-embedded (FFPE) samples from colorectal cancer patients and is available in the gene expression omnibus (GEO) repository under the accession numbers GSE86562 and GSE86559 for RNASeq and microarray data, respectively. The microarray gene expression data consists of 60,607 genes on 54 colorectal patients. We used the KRAS mutation status as a class variable. As a first step, the Affymetrix microarray data were log2-transformed and quantile-normalized, and genes with more than 50% missing values were filtered out. After that, we performed class comparison using the two-sample *t*-test at the 0.005 significant level threshold, which yielded 165 differentially expressed genes.

The RNASeq dataset contained 57,905 genes from the same colorectal cancer patients used to generate the microarray data. This data is in the form of counts, i.e., discrete. For this data, first, filtration was done to remove the genes with more than 50% of zeros across the samples, using the counts per million (CPM) method [31]. We retained genes whose CPM values are greater than 0.5. Thus, the dimension reduced to 17,473 genes. We performed differential expression analysis using the *DESeq2* package in *R*. This step reduced the genes to 282 genes using the 0.005 significance threshold level. The differential expression analysis tool in *DESeq2* uses a generalized linear model (GLM) of the following form:

$$\begin{aligned} g_{ij} &\sim NB(\mu_{ij}, \alpha_i) \\ \mu_{ij} &= s_j q_{ij} \\ log_2(q_{ij}) &= x_j.\beta_i, \end{aligned} \quad (1)$$

where g_{ij} is the counts for gene i in sample j. These counts are modeled using a negative binomial distribution with fitted mean μ_{ij} and a gene-specific dispersion parameter α_i. The fitted mean is decomposed into a sample-specific size factor s_j and a parameter q_{ij} proportional to the expected true concentration of fragments for sample j. The coefficients β_i represent the log2-fold changes for gene i for each column of the model or design matrix X. Note that the model can be generalized to use a sample- and gene-dependent normalization factors s_{ij}.

The dispersion parameter α_i defines the relationship between the variance of the observed count and its mean value. That is, how far we expect the observed count to be from the mean value, which depends both on the size factor s_j and the covariate-dependent part q_{ij} as defined above. Thus, the variance function is given by:

$$Var(g_{ij}) = E[(g_{ij} - \mu_{ij})^2] = \mu_{ij} + \alpha_i \mu_{ij}^2 \quad (2)$$

The steps performed by the *DESeq* function in *DESeq2* package are the estimation of s_j, and α_i, and fitting negative binomial GLM for β_i and Wald statistics by *nbinomWaldTest*.

We computed counts per million as:

$$CPM_i = \frac{g_i}{N} * 10^6, \quad (3)$$

where g_i denotes the counts observed from a gene of interest *i,* and *N* is the number of sequenced fragments.

RNASeq and microarray data integration may help improve cancer classification accuracy. Several studies have addressed the classification problem using RNASeq, microarray, or a combination of both, based on heterogeneous samples [20, 32, 33]. Our study aimed to integrate homogeneous samples from the RNASeq and microarray platforms. In this regard, we obtained the differentially expressed genes from the two platforms based on the same set of samples. After that, we used the database for annotation, visualization, and integrated discovery (DAVID) [34] and catalogue of somatic mutations in cancer (COSMIC) tools, to annotate the RNASeq transcripts list. The microarray genes symbol names were obtained from the dataset in [4]. We then obtained the intersection, complement of the intersection, and union between the two annotated lists.

Integration was done using the intersection, complement of the intersection, and the union of the two lists of genes. Due to the different nature of the two datasets, RNASeq was log2 transformed and quantile-normalized to make both types of data consistent with each other. Subsequently, the integration was done based on binding the two gene-lists from the RNASeq and microarray datasets. To transform the RNASeq data, we let:

$$\text{Transformed Data} = log_2(G + 1), \quad (4)$$

where *G* is the RNASeq counts data matrix, and *G + 1* is the RNASeq counts data matrix with all zero counts changed to one.

Quantile normalization ensures that probe intensities of each array in a set of arrays have the same distribution. A quantile-quantile plot would help to confirm if two probe vectors have the same distribution (quantiles lie on the diagonal line) or not. This approach can be extended to *n*-dimensional data. Let $q_k = (q_{k1}, \ldots, q_{kn})'$, *k= 1, ..., P*, be the vector of the k[th] quantiles for all *n* arrays, and $d = (\frac{1}{\sqrt{n}}, \ldots, \frac{1}{\sqrt{n}})'$ be the unit diagonal. To transform from the quantiles so that they all lie along the diagonal, we projected q on to d as below [35]:

$$\text{proj}_{\text{dqk}} = (\frac{1}{n}\sum_{j=1}^{n} \quad q_{1j}, \ldots, \frac{1}{n}\sum_{j=1}^{n} \quad q_{Pj}) \quad (5)$$

2.2. Data Integration

Here, we used homogeneous data from matched-pair samples from microarray and RNASeq technologies. Using a set-theoretic approach of taking the intersection, the complement of the intersection, or union, we obtained four lists of genes from the two platforms at the 0.005 significance level. The intersection between the two lists was 23 genes, with 401 genes being the complement of the intersection. The steps followed in this study are as shown at Fig. (**1**).

2.3. Classification Methods

Several methods have been developed for classification and their performance evaluated in both microarray and RNASeq platforms. Below, we briefly describe seven

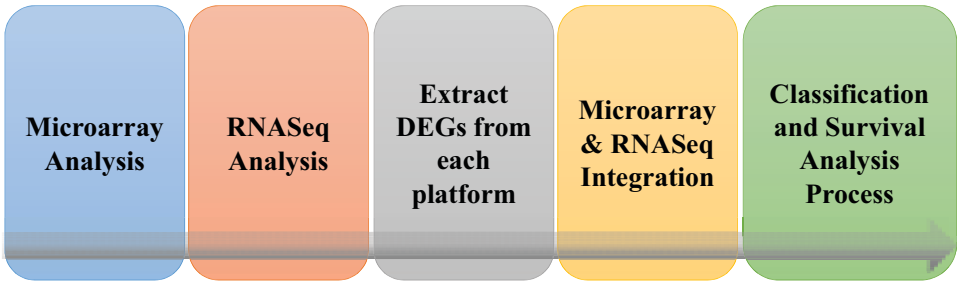


Fig. (1). Flow-chart of the analysis.

classification methods and how to evaluate their performances based on the integration of the two platforms.

2.3.1. Poisson Linear Discriminant Analysis

The PLDA classifier was proposed by Witten [36]. Witten used the Poisson log-linear model and developed an analog of diagonal linear discriminant analysis for sequence data.

Let G denote a $n * p$ matrix of read counts data, where n denotes the number of observations (samples), and p the number of genes. Let G_{ij} be the counts or reads for gene j in sample i; it is reasonable to assume that:

$$G_{ij} \sim Poisson(\mu_{ij}), \tag{6}$$

where $\mu_{ij} = s_i g_j$. To avoid identifiability issues, one can require $\sum_{i=1}^{n} s_i = 1$, where s_i is the number of counts per sample i, and g_j is the number of counts per gene j.

Suppose that we have K different classes of samples. Then we can write:

$$G_{ij}|y_i = k \sim Poisson(\mu_{ij} d_{kj}), \tag{7}$$

where y_i denotes the class of the i^{th} sample $(y_i = 1, 2, 3, \ldots, K)$ and d_{kj} denotes a measure of the level of the i^{th} gene to be differentially expressed in class k.

Let $g_i = (g_{i1}, g_{i2}, \cdots, g_{ip})'$ indicate the entries of row i in the G matrix, which are the gene expression levels of sample i. Let, $G_{.j} = \sum_{i=1}^{n} G_{ij}$, $G_{i.} = \sum_{j=1}^{p} G_{ij}$, and $G_{..} = \sum_{i,j} G_{ij}$ denote the column, row, and the overall totals, respectively. The maximum likelihood estimate (MLE) for μ_{ij} assuming independence is $\hat{\mu}_{ij} = \frac{G_{i.} G_{.j}}{G_{..}}$, and $\sum_{i=1}^{n} \hat{s}_i = 1$ yields the estimates $\hat{s}_i = \frac{G_{i.}}{G_{..}}$ and $\hat{g}_j = G_{.j}$. $\hat{s}_i$ is the estimate of the size factor for sample i. Maximum likelihood estimation provides the estimate of d_{kj} as $\hat{d}_{kj} = \frac{G_{c_k j}}{\sum_{i \in c_k} \hat{\mu}_{ij}}$, where c_k denotes the class of an observation.

If $\hat{d}_{kj} > 1$, then the j^{th} gene is overexpressed relative to the baseline in the k^{th} class, and if $\hat{d}_{kj} < 1$, then the j^{th} gene is under expressed relative to the baseline in the k^{th} class. If $G_{c_k j} = 0$ (an event that is not unlikely if the true mean for j^{th} gene is small), then the maximum likelihood estimates for d_{kj} equals zero.

Assume that we want to classify a new observation $g^* = (G_1^*, \cdots, G_p^*)$, and let y^* indicate the unknown class label. By Bayes rule,

$$P(y^* = k|g^*) \propto f_k(g^*)\pi_k, \tag{8}$$

where f_k is the density of a sample in class k and π_k is the prior probability that an observation belongs to class k. Then, if f_k is a normal density with a class-specific mean and common covariance, PLDA classifies a new sample to class k, which maximizes equation (**8**). Consequently, the discriminant score of PLDA is:

$$log\ Pr(y^* = k|g^*) \approx \sum_{j=1}^{P} G_j^* log\ d_{kj} - \sum_{j=1}^{P} s^* \lambda_j d_{kj} + log\ \pi_k + C \tag{9}$$

PLDA is implemented using the *R* package *MLSeq*.

2.3.2. Negative Binomial Linear Discriminant Analysis

Recently, Dong *et al.* [22] proposed NBLDA for RNASeq data analysis. NBLDA and Poisson linear discriminant analysis (PLDA) were considered the most suitable classifiers for RNASeq data due to the discrete nature of data [22, 36].

Let G_{ij} denote the number of reads in sample i, and gene j, $i = 1, 2, 3, \ldots, n$ and, $j = 1, 2, 3, \ldots, p$. Then G_{ij} is assumed to follow the negative binomial distribution:

$$G_{ij} \sim NB(\mu_{ij}, \phi_j), \mu_{ij} = s_i \lambda_j, \tag{10}$$

where s_i is the size factor, used to scale gene counts for the i^{th} sample due to different sequencing depth, λ_j is the total number of reads per gene, and $\phi_j \geq 0$ is the dispersion parameter. The mean and variance of the negative binomial distribution are given by:

$$E(G_{ij}) = \mu_{ij}$$
$$V(G_{ij}) = \mu_{ij} + \mu_{ij}^2 \phi_j. \tag{11}$$

Suppose that we have M classes. Let C_m be an indicator variable such that $C_m \in \{1, 2, 3, \cdots, M\}$. Then, the model for RNASeq data is:

$$(G_{ij}|y_i = m) \sim NB(\mu_{ij} d_{m,j}, \phi_j), \tag{12}$$

where d_{mj} denotes the differences among the M classes, and $y_i = m, m \in \{1, 2, 3, \cdots, M\}$ denotes the class of samples i. The assumption is that all the genes are independent.

Let $g^* = (G_1^*, \cdots, G_p^*)$ be a new sample whose class is to be predicted, s^* is the size factor, and y_i^* the class label value. By Bayes' rule, we have:

$$Pr(y^* = m|g^*g^*) \propto f_m(g^*g^*)\pi_m, \quad (13)$$

where f_m is the pdf of the sample in class m, and π_m is the prior probability that a sample comes from class m. The pdf of $G_{ij} = g_{ij}$ in equation (**12**) is:

$$Pr(G_{i,j} = g_{ij}|y_i = m) = \frac{\Gamma(g_{ij}+\phi_j^{-1})}{g_{ij}!\Gamma(\phi_j^{-1})}\left(\frac{s_i\lambda_j d_{mj}\phi_j}{1+s_i\lambda_j d_{mj}\phi_j}\right)^{g_{ij}}\left(\frac{1}{1+s_i\lambda_j d_{mj}\phi_j}\right)^{\phi_j^{-1}} \quad (14)$$

Thus, the discriminant score for NBLDA can be constructed from (**13**) and (**14**) as:

$$log\ Pr(y^* = m|g^*g^*) = \sum_{j=1}^{P} G_j^*[log\ d_{mj} - log\ (1 + s_i\lambda_j d_{mj}\phi_j)] - \sum_{j=1}^{P} \phi_j^{-1} log\ (1 + s_i\lambda_j d_{mj}\phi_j) + log\ \pi_m + C, \quad (15)$$

where C is a constant independent of m. The class m, which maximizes the score in equation (**15**) will be assigned to the new sample g^*g^*. NBLDA is implemented using the R package MLSeq.

2.3.3. Support Vector Machines

The SVM method was first proposed by Boser, Guyon, and Vapnik [37] at the Computational Learning Theory (COLT92) ACM Conference in 1992. The method is based on the idea of a hyperplane that lies furthermost from both classes. This plane is known as the *optimal (maximum) margin hyperplane*. The hyperplane is completely determined by a sub-set of the samples known as the *support vectors* [38]. SVM has the ability to handle problems where the data are not linearly separable by transforming the data using mapping kernel functions such as the radial basis function (RBF) kernel, polynomial function, and the linear function [39]. In addition, SVM can handle high dimensional data, which is an essential advantage in dealing with genetic data from cancer studies. This attribute makes SVM widely appealing and applicable to real-life data analysis problems such as handwritten character recognition, human face recognition, radar target identification, speech identification, and, quite recently, to gene expression data analysis [40, 41].

Suppose we have n samples and p genes. Further, assume samples belong to two distinct outcome classes represented by $+1$ or -1 and a feature vector g_i such that $(g_i, y_i) \in G \times Yi = 1,2, \ldots n$, where $g_i = (g_{i1}, g_{i2}, \ldots, g_{ip})'$ is the sample profile (vector) and $y_i \in \{+1, -1\}$ is the outcome class dichotomy. The goal is to classify the samples into one of the two classes by training the SVM which maps the input data (using a suitable kernel function) onto a high-dimensional space (feature space) $\{(\Phi(g_i), y_i)\}_{i=1}^n$. This is achieved by constructing an optimal separating hyperplane that lies furthest from both classes.

The general form of a separating hyperplane in the space of the mapped data is defined by:

$$\mathrm{w}^T\Phi(\mathrm{g}) + b = 0 \quad (16)$$

Here, $w = (w_1, w_2, \ldots, w_n)'$ is the weight vector. We can rescale the w and b such that the following equation determines the point in each class that is nearest to the hyperplane defined by the equation:

$$|\mathrm{w}^T\Phi(\mathrm{g}) + b| = 1 \quad (17)$$

Therefore, it should follow that for each sample i, $i \in \{1,2, \ldots, n\}$,

$$\boldsymbol{w}^T\Phi(g_i) + b = \{\geq 1 if y_i = +1 \ \leq -1 if y_i = -1 \quad (18)$$

After the rescaling, the distance from the nearest point in each class to the hyperplane becomes $\frac{1}{\|w\|}$. Thus, the distance between the two classes is $\frac{2}{\|w\|}$, which is called the *margin*. The solution of the following optimization problem is obtained to maximize the margin:

$$\min_{w,b} \|w\|^2$$
$$subject\ to$$
$$y_i(w^T\Phi(g_i) + b) \geq 1, i = 1,2, \ldots, n. \quad (19)$$

The square of the norm of w is considered to make the problem quadratic. Suppose w^* and b^* are the solutions to the optimization problem (**19**) above. Then this solution determines the hyperplane in the feature space where $(\mathrm{w}^*)^T\Phi(\mathrm{g}) + b^* = 0$. The points $\Phi(g_i)$ that satisfy the qualities $y_i((\mathrm{w}^{**})^T\Phi(\mathrm{g}_i) + b^*) = 1$ are called *support vectors* [38]. The SVM method is implemented using the *R* package *kernlab* [42].

2.3.4. Random Forests

Random forests were first introduced in 2001 [43, 44]. They are an extension of classification and regression trees, and also an improvement over bagged trees by further modification using a random small tweak to de-correlate the trees. Growing random forests leads to an improvement in prediction accuracy compared to single or bagged trees [45].

We build a number of forests of decision trees on bootstrapped training samples from the original data. A tree is obtained by recursively splitting the genes such that at each node of the tree, a candidate gene for splitting is obtained from a random sample of size v. A typical choice for v is such that $v \approx \sqrt{p}$, where p is the number of candidate genes for splitting.

We then grew the trees to maximum depth. Therefore, the two-step randomization process helps to de-correlate the trees [46]. To determine the prediction for an unknown sample, an average over all the trees is taken for a regression problem and a majority vote for a classification problem [43, 47, 48]. Random Forest Algorithm for Regression or Classification [43] can be implemented as follows

1. For $b = 1$ to B (# random-forest trees):
    - Draw a bootstrap sample of size N from the training data.
    - Grow a random-forest tree, T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the

tree, until the minimum node size, n_{min}, is reached.

- Select v genes at random from the p genes.
- Pick the best gene to split on among the v based on an impurity measure.
- Using the selected gene, split the node into two daughter nodes.

2. To predict a new sample x: Let $\hat{C}_b(x)$ be the class prediction of the b-th random-forest tree. Then $\hat{C}_{rf}^{B}(x) = majority\ vote\{\hat{C}_b(x)\}_{b=1}^{B}$

RF is implemented using the *R* package *randomForest* [49].

2.3.5. Artificial Neural Networks

Artificial neural networks (ANN) are multi-layered models that are constructed from three layers, each layer consisting of nodes called neurons [50]. The input layer contains nodes whose number is based on the input features. The output layer contains nodes equal to the number of classes, and finally, the hidden layer contains nodes determined by the level of tuning required. The inputs are weighted by multiplying each input by weight as a measure of its contribution. The layers are connected together *via* connection weights. These weights are determined through stages of model fitting. The hidden nodes receive the sum weighted from the input layer plus some bias. This summation is passed onto the transform function (activation function) to generate the results. These results are called outputs and interpreted as a class probability in our case.

There are many types of architecture of ANN. Neural networks are used widely in different fields, such as prediction in time series models, economic modeling, and medical applications [39]. Also, ANN can be applied to the classification problem using microarray gene expression data [50]. In this paper, we apply the method to both microarray and RNA Sequencing gene expression data.

Consider the simplest multi-layered network with one hidden layer. Assume we have gene expression data where n denotes the number of genes. Then the input layer receives the n gene expression levels for a sample, each multiplied by the corresponding weight, $w_{ij}^{(1)} g_j$, as shown in equation (**20**), below:

$$b_i = \sum_{j=0}^{n} \quad w_{ij}^{(1)} g_j \; i = 1,2,\dots,m, \tag{20}$$

where $g = (g_0, g_1, g_2, \dots, g_n)'$ is a vector of input features and $g_0 = 1$ is a constant input feature that with weight w_{i0}. The quantities, b_i, are called *activations*, and the parameters $w_{ij}^{(1)}$ are the weights. Note that alternatively b_i can be viewed as a summary of the n genes from sample i. The superscript "(1)" indicates that this is the first layer of the network. Each of the activations is then transformed by a nonlinear activation function f, typically a sigmoid, as in equation (**21**) below:

$$z_i = f(b_i) = \frac{1}{1+exp\,(-b_i)} \tag{21}$$

The quantities z_i are interpreted as the output of hidden units, so-called because they do not have values specified by the problem (as is the case for input units) or target values used in training (as is the case for output units).

In the second layer, the outputs of the hidden units are linearly combined to give the activations:

$$a_k = \sum_{i=0}^{m} \quad w_{ik}^{(2)} z_i \; k = 1,2,\dots,K \tag{22}$$

Again, $z_0 = 1$ corresponds to the bias. Weights $w_{ik}^{(2)}$ parameterize the transformations in the second layer of the neural network. The output units are transformed using an activation function. Again, a sigmoid function may be used as shown below:

$$y_k = f(a_k) = \frac{1}{1+exp\,(-a_k)} \tag{23}$$

These equations may be combined to give the overall equation describing the forward propagation through the network, and describes how an output vector is computed from an input vector, given the weight matrices as:

$$y_k = f(\sum_{i=0}^{m} \quad w_{ik}^{(2)} f(\sum_{j=0}^{n} \quad w_{ij}^{(1)} g_i)) \tag{24}$$

ANN is implemented using the *R* package *nnet* [51].

2.3.6. Naïve Bayes

The Naive Bayes classifier uses probability theory to find the most likely of the possible classes in a classification problem. The NB classifier relies on two assumptions, namely, that each attribute is conditionally independent of the other attributes given the class and that all the attributes have an influence on the class [52]. The popularity of this classifier is mainly due to its simplicity, yet exhibiting a surprisingly competitive predictive accuracy. The NB classifier has previously been applied in many fields, including microarray gene expression data [39, 50].

Consider an $n * p$ gene expression data matrix, where n is the number of the samples, and p is the number of the genes (features). Let $g_{kj}, j = 1,2,\dots,p$, denote the j^{th} gene on the k^{th} sample. Let C_i be the i^{th} class, $i = 1,2,3,\dots,L$. The Naive Bayes classifier uses the *maximum a posteriori* (MAP) classification rule to classify these samples. The probability of the k^{th} sample gene information vector, $\boldsymbol{G}_k = (g_{k1}, g_{k2}, \dots, g_{kp})'$, is calculated, and then the sample is assigned the class with the largest probability from L conditional probabilities.

Let $P(C_1|G_k), P(C_2|G_k), \dots, P(C_L|G_k)$ denote the set of L conditional probabilities. The NB classification depends on the Bayes rule, which states that a posterior probability:

$$P(\boldsymbol{G}_k) = \frac{P(C_i)P(C_i)}{P(\boldsymbol{G}_k)} \propto P(C_i)P(C_i), k = 1,2,\dots,n \tag{25}$$

where $P(G_k)$ is considered a common normalizing factor for all the L probabilities.

The NB classification assumes that all input features are conditionally independent, that is,

Table 1. The number of genes obtained through the intersection, the complement of the intersection, and union of the gene-lists from differential expression analysis (RNASeq: GSE86562, Microarray: GSE86559).

Dataset	Total of DEGs	Intersection	Complement of Intersection	Union
GSE86559	165	23	142	424
GSE86562	282		259	

$$P(g_{k1}, g_{k2}, \dots, g_{kp}|C_i) =$$
$$P(g_{k1}|g_{k2}, \dots, g_{kp}, C_i)P(g_{k2}, \dots, g_{kp}|C_i) =$$
$$P(g_{k1}|C_i)P(g_{k2}, \dots, g_{kp}|C_i) =$$
$$P(g_{k1}|C_i)P(g_{k2}|C_i) \dots P(g_{kp}|C_i). \quad (26)$$

Ultimately, NB classifies a new sample, G^*, according to the model with MAP probability given the sample, as:

$$Class(G^*)_{MAP} = argmax(P(C_i|\boldsymbol{G}^*)) \quad (27)$$

NB is implemented using the *R* package *naivebayes*.

2.3.7. k-Nearest Neighbors

The *k*-nearest neighbor classifiers (KNN) are known to be the most useful instance-based learners. KNN is a non-parametric model [51]. If the classification is based on Euclidean distance in a feature space, then *k* determines the number of neighbors to be used. In the testing set, the new sample is assigned to the class that is most likely among the *k* neighbors. Then the number of neighbors can be tuned to choose the optimal fitted model parameters [39, 50].

The KNN uses the Euclidean distance measure to find the closest samples for the new sample. Suppose we have two samples, each one with n genes. Denote the two samples as $S_1 = (g_{11}, g_{12}, \dots, g_{1n})'$ and $S_2 = (g_{21}, g_{22}, \dots, g_{2n})'$. Then the Euclidean distance is calculated as the square root of the sum of the squared differences in their corresponding values. Using the Euclidean distance formula, the distance between two points, $dist(S_1, S_2)$, is given as:

$$dist(S_1, S_2) = \sqrt{\sum_{j=1}^{n} (g_{1j} - g_{2j})^2} \quad (28)$$

where a large $dist(S_1, S_2)$ means the two samples belong to different classes, and values near zero suggest that the samples are homogeneous. KNN is implemented using the *R* package *caret*.

3. RESULTS

The analysis of RNASeq data using the integrated list of genes was performed using R statistical software. Assessment of the methods was done using 10-fold cross-validation. Here, the 54 CRC samples were divided into 10-folds randomly, with each fold consisting of about 5 - 6 samples. After that, we used a nine-folds for model-building and one-fold for the testing and validation. Thus, this process was self-iterated ten times, and the average of the ten iterations used to obtain the model performance measures. Several performance measures exist in the literature that can assess classification based on microarray and RNASeq gene expression data. The metrics include accuracy, sensitivity, specificity, kappa coefficient, AUC, and balanced error rate (BER) [54, 55].

Table **1** provides the number of genes obtained through the intersection, complement of the intersection, and union of the gene-lists from differential expression analysis (RNASeq: GSE86562, Microarray: GSE86559). There were 165 and 282 total DEGs in the GSE86559 and GSE86562 datasets, respectively. We obtained 23 genes through the intersection, 142 from a complement of GSE86559, 259 a complement of GSE86562, and 424 from a union (Table **1**).

The 23 genes obtained from the intersection of the RNASeq and microarray gene expression data, their official gene symbols, and names are in Table **2**.

We performed an exploratory analysis of the RNASeq data. Fig. (**2**) shows the most meaningful changes at the 0.005 significance level among the genes between the two conditions, based on the volcano plot [56]. The volcano plot shows the genes with smaller p-values (higher $-log_{10}$ values) in red.

Fig. (**3**) illustrates the estimated dispersion of the RNASeq data using the *DESeq2* package, with each gene having a gene-specific dispersion parameter. Good estimates of dispersion parameters lead to accurate detection of differen-tially expressed genes. Underestimating the dispersion parameters might lead to false positives (*i.e.*, declaring genes to be differentially expressed when they are not truly differentially-expressed). On the other hand, overestimating the dispersion parameters might lead to false negatives [57].

Tables **3-6** show the performance of the gene-lists in predicting mutation status, based on seven methods (algorithms), at the 0.005 significance level: the 282 gene-list (Table **3**); the 23 gene-list (Table **4**); the 424 gene-list (Table **5**); and the 401 gene-list (Table **6**).

It is apparent from Table **3**, compared to Table **4** below that NB, ANN, KNN, and PLDA were improved in the common 23 genes in terms of all performance measures, while RF and NBLDA had the same performance. SVM had a better result on the full list of 282 genes. Therefore, in general, four methods out of seven were improved on the 23 gene-list compared to the 282 genes-list. From Fig. (**4a** and **b**), we notice NBLDA works very well in both lists of genes.

Table 2. The official gene symbols and the corresponding gene names.

Ensemble Gene ID	Official Gene Symbol	Name
ENSG00000108511	HOXB6	Homeobox B6(HOXB6)
ENSG00000169247	SH3TC2	SH3 domain and tetratricopeptide repeats 2(SH3TC2)
ENSG00000120068	HOXB8	Homeobox B8(HOXB8)
ENSG00000025293	PHF20	PHD finger protein 20(PHF20)
ENSG00000136997	MYC	v-myc avian myelocytomatosis viral oncogene homolog (MYC)
ENSG00000143882	ATP6V1C2	ATPase H+ transporting V1 subunit C2(ATP6V1C2)
ENSG00000003096	KLHL13	Kelch like family member 13(KLHL13)
ENSG00000131746	TNS4	Tensin 4(TNS4)
ENSG00000196532	HIST1H3C	Histone cluster 1 H3 family member c(HIST1H3C)
ENSG00000233101	HOXB-AS3	HOXB cluster antisense RNA 3(HOXB-AS3)
ENSG00000204104	TRAF3IP1	TRAF3 interacting protein 1(TRAF3IP1)
ENSG00000126003	PLAGL2	PLAG1 like zinc finger 2(PLAGL2)
ENSG00000120875	DUSP4	Dual specificity phosphatase 4(DUSP4)
ENSG00000164070	HSPA4L	Heat shock protein family A (Hsp70) member 4 like (HSPA4L)
ENSG00000111057	KRT18	Keratin 18(KRT18)
ENSG00000260807	LMF1	Lipase maturation factor 1(LMF1)
ENSG00000174136	RGMB	Repulsive guidance molecule family member b(RGMB)
ENSG00000197818	SLC9A8	Solute carrier family 9 member A8(SLC9A8)
ENSG00000187372	PCDHB13	Protocadherin beta 13(PCDHB13)
ENSG00000140526	ABHD2	Abhydrolase domain containing 2(ABHD2)
ENSG00000166068	SPRED1	Sprouty related EVH1 domain containing 1(SPRED1)
ENSG00000182742	HOXB4	Homeobox B4(HOXB4)
ENSG00000101193	GID8	GID complex subunit 8 homolog (GID8)

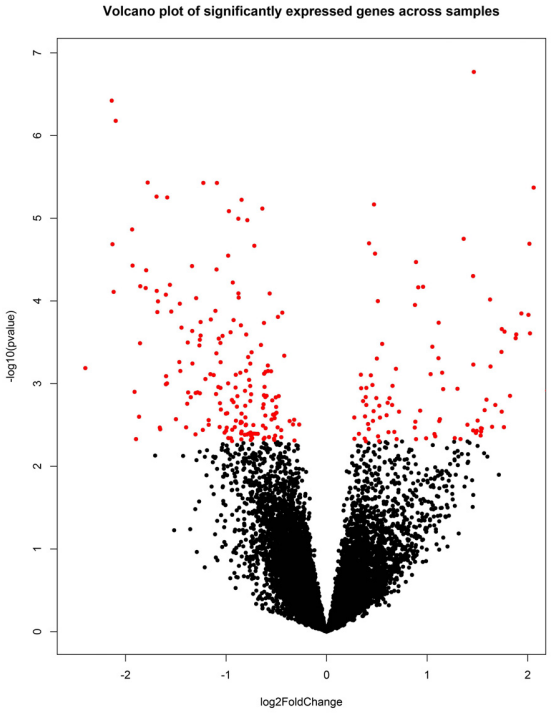


Fig. (2). Volcano plot of the RNASeq dataset shows the 282 differentially expressed genes in red points (α = 0:005).

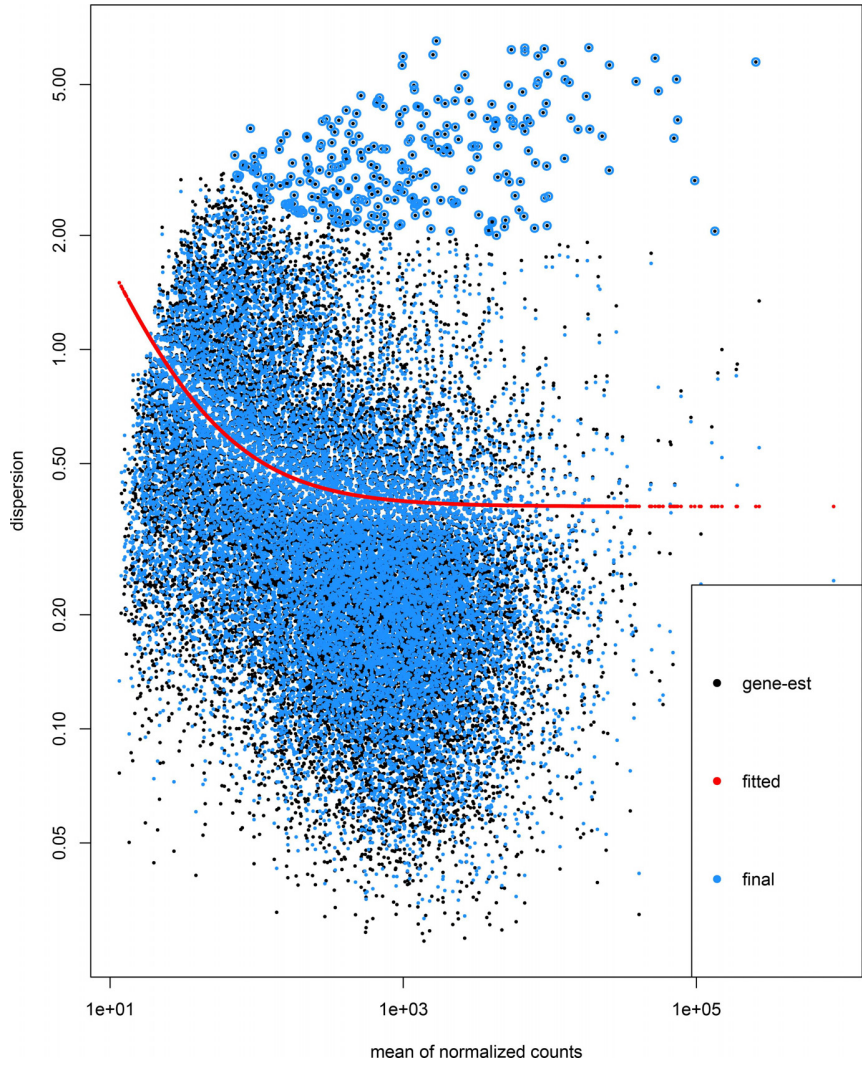


Fig. (3). Dispersion for the RNASeq data.

Table 3. Performance of the classification methods for the 282 gene-list, on the RNASeq dataset ($\alpha = 0.005$).

Metric	Methods						
	SVM	**NB**	**RF**	**ANN**	**KNN**	**NBLDA**	**PLDA**
Accuracy (95% CI)	0.80 (0.66, 0.89)	0.76 (0.62, 0.87)	0.83 (0.71, 0.92)	0.72 (0.58, 0.84)	0.72 (0.58, 0.84)	0.89 (0.77, 0.96)	0.80 (0.66, 0.89)
Sensitivity (95% CI)	0.89 (0.71, 0.98)	0.59 (0.39, 0.78)	0.78 (0.58, 0.91)	0.78 (0.58, 0.91)	0.67 (0.46, 0.83)	0.81 (0.62, 0.94)	0.81 (0.62, 0.94)
Specificity (95% CI)	0.70 (0.50, 0.86)	0.93 (0.76, 0.99)	0.89 (0.71, 0.98)	0.67 0.46, 0.83)	0.78 (0.58, 0.91)	0.96 (0.81, 1.00)	0.78 (0.58, 0.91)
Kappa (95% CI)	0.59 (0.38, 0.80)	0.52 (0.30, 0.73)	0.67 (0.47, 0.86)	0.44 (0.21, 0.68)	0.44 (0.21, 0.68)	0.78 (0.61, 0.94)	0.59 (0.38, 0.81)
AUC	0.86	0.77	0.87	0.72	0.78	0.94	0.80
BER	0.19	0.21	0.16	0.28	0.28	0.10	0.20

Table 4. Performance of the classification methods for the 23 gene-list, on the RNASeq dataset ($\alpha = 0.005$).

Metric	Methods						
	SVM	NB	RF	ANN	KNN	NBLDA	PLDA
Accuracy (95% CI)	0.78 (0.64, 0.88)	0.80 (0.66, 0.89)	0.83 (0.71, 0.92)	0.80 (0.66, 0.89)	0.76 (0.62, 0.87)	0.89 (0.77, 0.96)	0.87 (0.75, 0.95)
Sensitivity (95% CI)	0.81 (0.62, 0.94)	0.70 (0.50, 0.86)	0.78 (0.58, 0.91)	0.81 (0.62, 0.94)	0.70 (0.50, 0.86)	0.85 (0.66, 0.96)	0.85 (0.66, 0.96)
Specificity (95% CI)	0.74 (0.54, 0.89)	0.89 (0.71, 0.98)	0.89 (0.71, 0.98)	0.78 (0.58, 0.91)	0.81 (0.62, 0.94)	0.93 (0.76, 0.99)	0.89 (0.71, 0.98)
Kappa (95% CI)	0.56 (0.33, 0.78)	0.59 (0.38, 0.80)	0.67 (0.47, 0.86)	0.59 (0.38, 0.81)	0.52 (0.29, 0.75)	0.78 (0.61, 0.94)	0.74 (0.56, 0.92)
AUC	0.80	0.82	0.91	0.84	0.78	0.89	0.91
BER	0.22	0.19	0.16	0.20	0.24	0.11	0.13

Table 5. Performance of the classification methods for the 424 gene-list, on the combined RNASeq and microarray datasets ($\alpha = 0.005$).

Metric	Methods				
	SVM	NB	RF	ANN	KNN
Accuracy (95% CI)	0.98 (0.90, 1.00)	0.93 (0.82, 0.98)	0.93 (0.82, 0.98)	0.98 (0.90, 1.00)	0.83 (0.71, 0.92)
Sensitivity (95% CI)	1.00 (0.87, 1.00)	0.89 (0.71, 0.98)	0.89 (0.71, 0.98)	1.00 (0.87, 1.00)	0.74 (0.54, 0.89)
Specificity (95% CI)	0.96 (0.81, 1.00)	0.96 (0.81, 1.00)	0.96 (0.81, 1.00)	0.96 (0.81, 1.00)	0.93 (0.76, 0.99)
Kappa (95% CI)	0.96 (0.89, 1.00)	0.85 (0.71, 0.99)	0.85 (0.71, 0.99)	0.96 (0.89, 1.00)	0.67 (0.47, 0.86)
AUC	1.00	0.94	0.96	1.00	0.89
BER	0.02	0.07	0.07	0.02	0.15

Table 6. Performance of the methods for the 401 gene-list, on the RNASeq dataset ($\alpha = 0.005$).

Metric	Methods				
	SVM	NB	RF	ANN	KNN
Accuracy (95% CI)	0.98 (0.90, 1.00)	0.93 (0.82, 0.98)	0.93 (0.82, 0.98)	0.96 (0.87, 1.00)	0.83 (0.71, 0.92)
Sensitivity (95% CI)	1.00 (0.87, 1.00)	0.89 (0.71, 0.98)	0.89 (0.71, 0.98)	0.96 (0.81, 1.00)	0.74 (0.54, 0.89)
Specificity (95% CI)	0.96 (0.81, 1.00)	0.96 (0.81, 1.00)	0.96 (0.81, 1.00)	0.96 (0.81, 1.00)	0.93 (0.76, 0.99)
Kappa (95% CI)	0.96 (0.89, 1.00)	0.85 (0.71, 0.99)	0.85 (0.71, 0.99)	0.93 (0.83, 1.00)	0.67 (0.47, 0.86)
AUC	1.00	0.91	0.96	1.00	0.89
BER	0.02	0.07	0.07	0.04	0.15

Table **5** presents the integration results using the union approach, and it is clear that SVM, NB, RF, ANN, and KNN methods were improved compared to the case of 282 differentially expressed genes. Fig. (**4a** and **c**) confirm these results. Moreover, SVM and ANN had a higher accuracy than the other methods.

As can be seen from Table **6** above, the methods performed better for the gene-list of 401 genes, compared to the 282 gene-list. Furthermore, Fig. (**4d**) confirm these results.

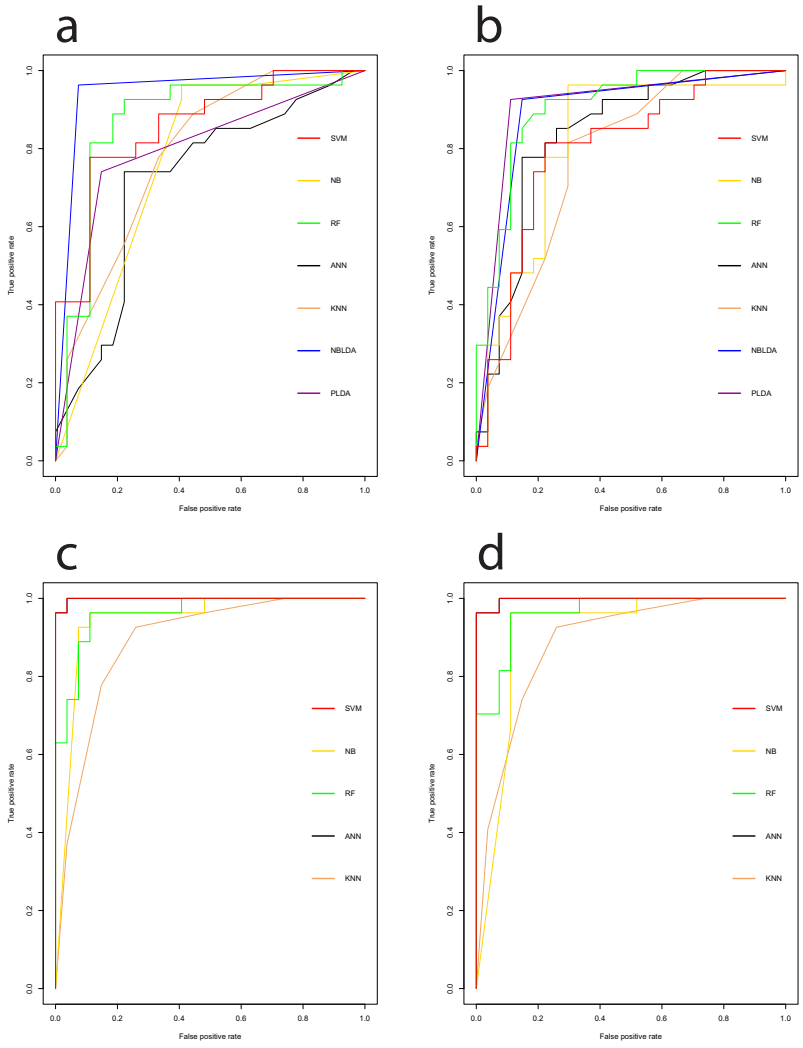


Fig. (4). ROC curves based on the (**a**) 282 gene-list for the RNASeq data, (**b**) 23 gene-list for the RNASeq data, (**c**) 424 gene-list for the RNASeq and microarray datasets, and (**d**) 401 gene-list for the RNASeq and microarray datasets, under (α = 0:005).

We compared our gene-list of 23 genes with the 18-gene RAS signature (DUSP4, DUSP6, ELF1, ETV4, ETV5, FXYD5, KANK1, LGALS3, LZTS1, MAP2K3, PHLDA1, PROS1, S100A6, SERPINB1, SLCO4A, SPRY2, TRIB2, and ZFP106) as reported by Dry *et al.* [58] and found only one overlapping gene (DUSP4). It turned out that this was also the most predictive of the seven genes (DUSP4, DUSP6, ETV4, ETV5, PHLDA1, SERPINB1, and TRIB2) that were discussed in Omolo *et al.* (2016) [4].

We performed an additional analysis to assess whether the 23 gene-list was predictive of overall survival (OS). We used the mutation status as a group variable and vital status (dead or alive) as the censoring variable in this analysis. Overall, there were 20 deaths out of the 54 samples. The results shows that the median OS was 1692 days for the 54 samples. We used the Kaplan-Meier curves to graphically compare survival probabilities (Fig. **5**) between the two mutation groups (RAS-mutant *vs*. wild-type), and the log-rank test using the RAS mutation status as the group variable. There no significant difference in OS between the two groups (log-rank = 1.8, p-value = 0.2). We then applied the Cox proportional hazards (CPH) model was to assess the significance of the 23 genes and RAS mutation status. The results show that 9 of 23 genes were significantly associated with OS, including *SPRED1, KLHL13, HOXB4, LMF1, HSPA4L* at the 0.05 level, and *ATP6V1C2, PLAGL2, MYC, SLC9A8* at the 0.1 level (LRT = 56.85, p-value = 0.0002) as can be seen in Table **7**.

We further performed an analysis of the top nine genes using gradient boosted trees and Shapley additive explanations (SHAP) methods to identify the top-K genes (*1* < *K* < *9*) [59]. The SHAP approach determined the order of importance of our nine genes. SHAP values give the importance of a gene by comparing what a model predicts with and without the gene. A SHAP value of 0 means that the gene does not affect the prediction, as shown in Fig. (**6**). The vertical axis shows the gene names, arranged in the order of importance, from top to bottom while the adjacent value next to the gene name is the mean SHAP value. The horizontal axis shows the SHAP value, which indicates how

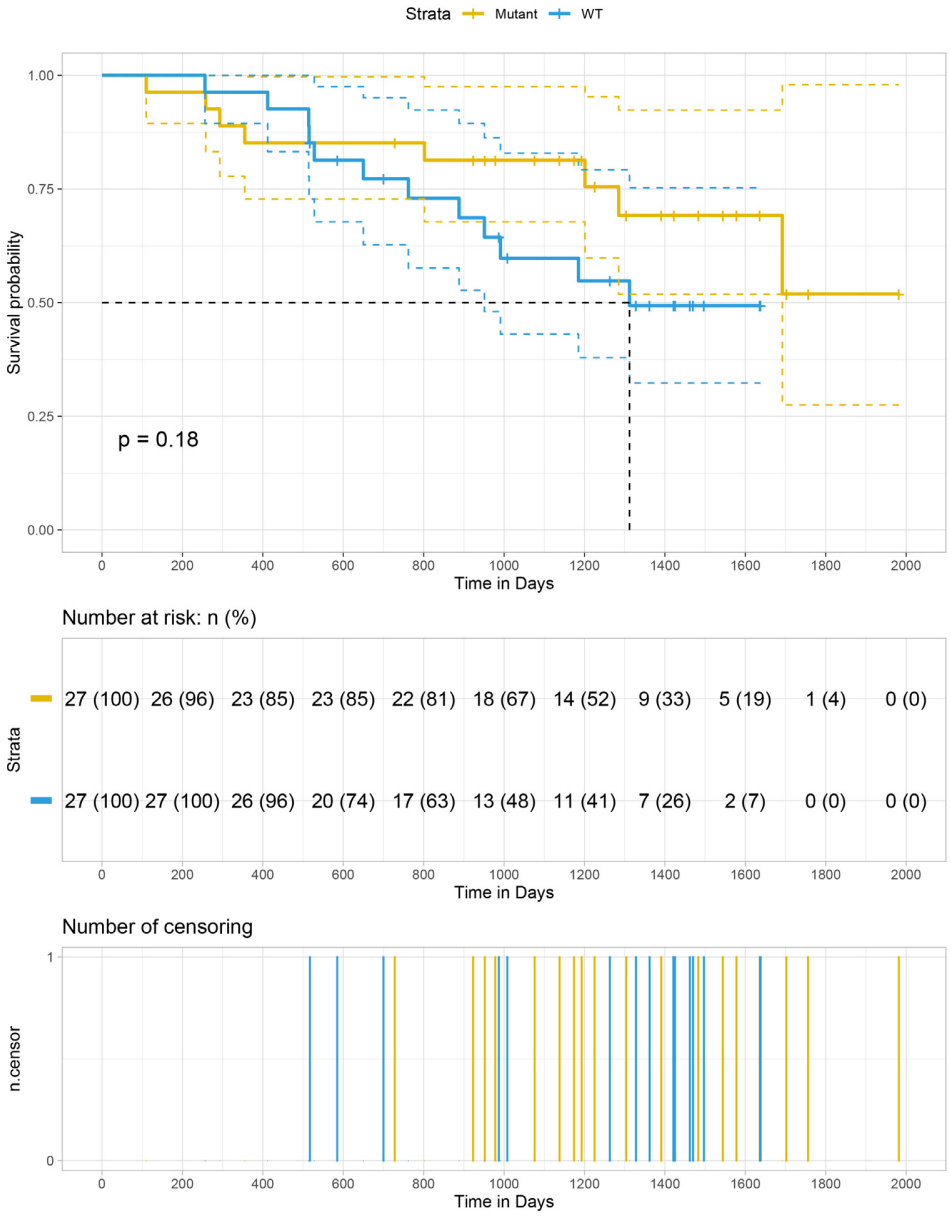


Fig. (5). Kaplan-Meier curves for overall survival (in months).

much the change was in log-odds. From the log-odds, one can obtain the probability of success. The gradient color indicated the original value for that gene. Genes pushing the prediction higher are colored blue, while those pushing the prediction lower are colored yellow. Each point represents a row from the original dataset.

4. DISCUSSION

The development of molecular signatures is a significant step towards understanding the molecular mechanisms of tumor genesis, which could help with accurate prognosis and diagnosis and thus allow physicians to prescribe suitable patient-specific therapies.

Several studies have done cancer classification using either microarray or RNASeq data only, and few have shown integration of both types of data, based on heterogeneous datasets. To the best of our knowledge, no cancer classification study has employed the integration of a homogeneous datasets approach. In this study, we integrated homogeneous microarray and RNASeq datasets and assessed whether such an approach could improve the classification accuracy using seven methods, namely, SVM with radial basis function kernel, NB, RF, ANN, KNN, NBLDA, and PLDA. We implemented the classification of the mutation status of CRC samples, using gene-lists obtained through the intersection, the complement of an intersection, and the union of differentially-expressed genes from microarray and RNASeq datasets.

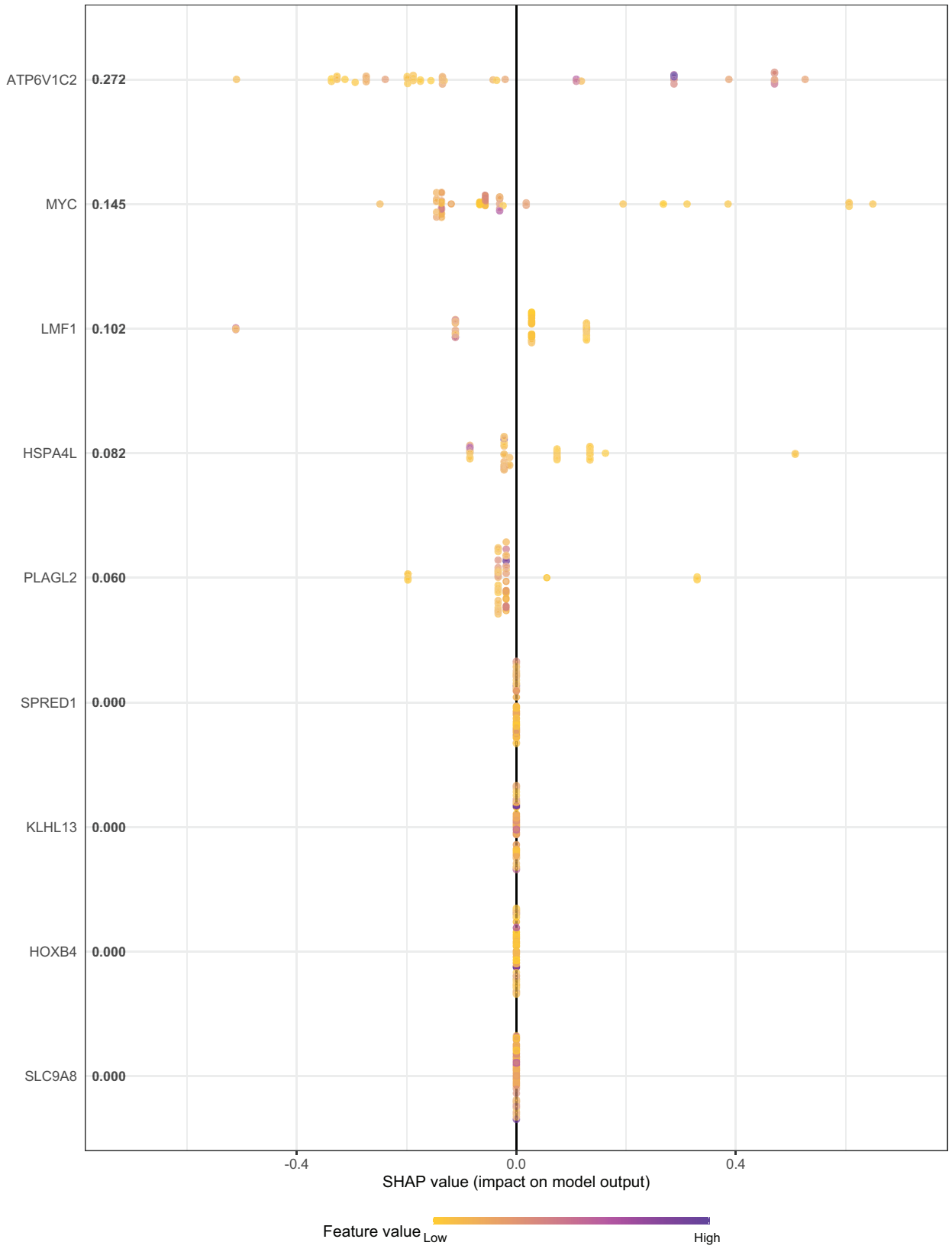


Fig. (6). Genes in ascending order of importance (Note: dots represent SHAP values of specific features).

Table 7. Cox proportional hazards model for overall survival, using the 23 genes and RAS mutation status (class) as covariates.

Covariate	Coef	Hazard Ratio (HR)	SE(Coef)	Z-score	P-value
Class	2.23E+00	9.34E+00	1.41E+00	1.589	0.112
ATP6V1C2	4.72E-03	1.01E+00	2.66E-03	1.779	0.0752 .
HOXB-AS3	7.96E-05	1.00E+00	1.30E-03	0.061	0.951
KRT18	-3.27E-05	1.00E+00	8.73E-05	-0.374	0.7084
RGMB	7.48E-04	1.00E+00	1.13E-03	0.662	0.5079
PLAGL2	-3.28E-03	9.97E-01	1.89E-03	-1.737	0.0824 .
DUSP4	2.12E-03	1.00E+00	2.24E-03	0.946	0.3441
SPRED1	1.50E-03	1.00E+00	7.61E-04	1.969	0.0489 *
SH3TC2	6.92E-04	1.00E+00	4.42E-04	1.564	0.1178
HOXB8	-1.04E-02	9.90E-01	7.21E-03	-1.444	0.1488
ABHD2	4.67E-04	1.00E+00	4.34E-04	1.078	0.2812
TNS4	-5.58E-04	9.99E-01	3.74E-04	-1.491	0.1358
HIST1H3C	-4.67E-03	9.95E-01	3.20E-03	-1.458	0.1449
KLHL13	8.28E-03	1.01E+00	3.24E-03	2.559	0.0105 *
MYC	1.21E-03	1.00E+00	7.16E-04	1.689	0.0911 .
HOXB4	9.10E-03	1.01E+00	3.60E-03	2.529	0.0114 *
HOXB6	-2.78E-04	1.00E+00	4.18E-03	-0.067	0.9469
PHF20	1.47E-03	1.00E+00	1.78E-03	0.825	0.4094
LMF1	3.19E-03	1.00E+00	1.43E-03	2.224	0.0262 *
SLC9A8	-4.98E-03	9.95E-01	2.56E-03	-1.946	0.0517 .
GID8	2.16E-03	1.00E+00	2.61E-03	0.828	0.4079
HSPA4L	-6.94E-03	9.93E-01	2.78E-03	-2.497	0.0125 *
PCDHB13	-3.90E-03	9.96E-01	4.09E-03	-0.953	0.3406
TRAF3IP1	-7.63E-03	9.92E-01	4.86E-03	-1.571	0.1163
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					

CRC is the third most common cancer and one of the leading causes of death around the world. The findings suggest that combining two homogeneous datasets from different technologies could lead to an increase in CRC classification accuracy. Castillo *et al.* [20] reported that combining heterogeneous datasets from different platforms can improve the performance of a classifier, using multiple datasets. They used data from different technologies and platforms to obtain a larger sample size due to the lack of enough RNASeq samples. Our proposed approach is different from Castillo *et al.* [20], in that we used homogeneous datasets and a balanced binary class problem. We used the 0.005 significance level to obtain the differentially expressed genes, which is restrictive enough to control the false positive rate.

A comparison of the performance of the classification methods for each gene-list revealed that SVM yielded the highest mean accuracy (0.885), followed by RF (0.880), ANN(0.865), NB(0.855), and KNN(0.785) across the four gene-lists. However, NBLDA performed better than PLDA as a classifier when the analysis was restricted to RNASeq (count) data. Castillo *et al.* [20] also showed that SVM performed second to KNN. Statnikov *et al.* [60] performed a comparison of 18 classification methods on five feature selection methods, using eight datasets and showed that RF had the highest accuracy (0.954). Our classification results using the integrated list of genes outperformed *Mamatjan et al.* [61], where they used RNASeq data for tumor classification. Their results showed that mRNA signatures and DNA methylation signatures as single platforms achieved 95% and 88% accuracy of histological diagnosis, respectively. Moreover, the *PLAGL2* gene, as one of our predictive genes, also has been one of the most common predictive genes in *Rashid et al.* [62]. Furthermore, our work integrated RNA and DNA signature for classification and survival analysis with very high metrics compared to *Popovici et al.* [63]. In this study [63], the authors developed

a classifier using 64 genes for detecting BRAF mutant tumors for colon cancer. Also, they found *DUSP4* to be one of the top 50 differently expressed genes.

Survival analysis results showed that 9 of 23 genes were prognostic for overall survival for CRC patients. Upon subjecting the nine genes to the Shapley additive explanations (SHAP) method to rank the genes in order of importance, the top-5 genes to emerge were *ATP6V1C2, MYC, LMF1, HSPA4L,* and *PLAGL2*.

Our findings are consistent with other published molecular signatures from previous studies [64, 65, 66]. Zumwalt *et al.* [64] showed that *ATP6V1C2* expression successfully distinguished between cancerous and non-cancerous samples in CRC. He *et al.* [65] reported that the expression of ***c-Myc***, which was one of the three related human genes encoded under ***MYC*** genes family, was observed in many human cancers and was elevated in up to 70 - 80% in CRC. Liu *et al.* [66] identified ten lncRNAs related to crucial outcomes in CRC, and one of these was ***LMF1***. Zhang *et al.* [67] obtained 34 genes using minimal redundancy maximal relevance (mRMR) and incremental feature selection (IFS) methods. They found that the ***HSPA4L*** gene was the most highly expressed in CRC patients with chromosomal instability (CIN) mechanism. Zheng *et al.* [68] reported that the ***PLAGL2*** gene was vital in increasing the effect on glioblastoma and colorectal cancer. Su *et al.* [69] reported that ***PLAGL2*** served as an oncogenic function in multiple human malignancies, including colorectal cancer (CRC).

This study was limited by the available number of homogeneous RNASeq and microarray datasets. Only one matched-pair set of 54 CRC samples was analyzed. Future studies should extend the approach to more than one cancer type and multiple datasets. However, the number of samples in each dataset (n = 54) ensured that the training and validation sets were large enough for the magnitude and statistical significance of the classification accuracies.

CONCLUSION

In summary, data integration by taking the *intersection* of the individual gene-lists from the two data types improved the classification accuracy of CRC. However, laboratory experiments should be conducted on this 23-gene signature to further assess its clinical significance in CRC research. NBLDA method was the best performer on the RNASeq data. Results suggest that the SVM method was the most suitable classifier for CRC across the two data types and had high accuracy before and after the integration. Future studies should determine the effectiveness of integration in cancer survival analysis and the application on unbalanced data (where the classes are of different sizes) as well as on data with multiple classes.

LIST OF ABBREVIATIONS

CRC	=	Colorectal Cancer
SSA	=	Sub-Saharan Africa
EGFRi	=	Anti-epithelial Growth Factor Receptor Inhibitor
FFPE	=	Formalin-Fixed Paraffin-Embedded
AUC	=	Area Under the ROC Curve
WHO	=	World Health Organization
NGS	=	Next-Generation Sequencing
PLDA	=	Poison Linear Discriminant Analysis
NBLDA	=	Negative Binomial Linear Discriminant Analysis
SVM	=	Support Vector Machines
ANN	=	Artificial Neural Networks
LDA	=	Linear Discriminant Analysis
RF	=	Random Forests
NB	=	Naive Bayes
KNN	=	k-Nearest Neighbors
IG	=	Information Gain
SGA	=	Standard Genetic Algorithm
CFS	=	Correlation-based Feature Selection
IBPSO	=	Improved-Binary Particle Swarm Optimization
bagSVM	=	Bagging SVM
CART	=	Classification and Regression Trees
mRMR	=	Minimum-redundancy Maximum-Relevance
GEO	=	Gene Expression Omnibus
CPM	=	Counts Per Million
GLM	=	Generalized Linear Model
DAVID	=	Database for Annotation, Visualization, and Integrated Discovery
COSMIC	=	Catalogue of Somatic Mutations in Cancer
MLE	=	Maximum Likelihood Estimate
RBF	=	Radial Basis Function
MAP	=	Maximum a Posteriori
BER	=	Balanced Error Rate
TP	=	True Positive
TN	=	True Negative
FP	=	False Positive
FN	=	False Negative
RA	=	Random Accuracy
OS	=	Overall Survival
CPH	=	Cox Proportional Hazards
SHAP	=	Shapley Additive Explanations
IFS	=	Incremental Feature Selection

CIN = Chromosomal Instability

HR = Hazard Ratio

AUTHORS' CONTRIBUTION

BO conceived the study. MM performed all the analyses. MM and BO drafted the manuscript. MM, HM, and BO proof-read, discussed, and approved the final manuscript.

ETHICS APPROVAL AND CONSENT TO PARTICIPATE

Not applicable.

HUMAN AND ANIMAL RIGHTS

No animals/humans were used for studies that are basis of this research.

CONSENT FOR PUBLICATION

Not applicable.

AVAILABILITY OF DATA AND MATERIALS

The datasets supporting the results of this article have been deposited to the public repository (GEO) under the series accession number GSE86566 https: //www.ncbi.nlm.nih.gov/geo/. The individual datasets are accessible under the accession number GSE86559 for the microarray data and GSE86562 for the RNASeq data. The mutation and overall survival data are available upon request from BO.

FUNDING

This work was funded by GSK Africa Non-Communicable Disease Open Lab through the DELTAS Africa Sub-Saharan African Consortium for Advanced Biostatistics (SSACAB) Grant No. 107754/Z/15/Z- training programme. The views expressed in this publication are those of the author(s) and not necessarily those of GSK.

CONFLICT OF INTEREST

The authors declare no conflict of interest, financial or otherwise.

ACKNOWLEDGEMENTS

The authors wish to thank Prof. Bob Gagnon (GlaxoSmithKline (GSK)).

REFERENCES

[1] Gandomani HS, Yousefi SM, Aghajani M, *et al.* Colorectal cancer in the world: incidence, mortality and risk factors. Biomed Res Ther 2017; 4(10): 1656-75. http://dx.doi.org/10.15419/bmrat.v4i10.372

[2] Siegel RL, Miller KD, Jemal A. Cancer statistics, 2019. CA Cancer J Clin 2019; 69(1): 7-34. http://dx.doi.org/10.3322/caac.21551 PMID: 30620402

[3] Word Cancer Research Fund. colorectal cancer statistics. Available at: www.wcrf.org/dietandcancer/cancer-trends/colorectal-cancer-statistics. (Accessed on August 2, 2019).

[4] Omolo B, Yang M, Lo FY, *et al.* Adaptation of a RAS pathway activation signature from FF to FFPE tissues in colorectal cancer. BMC Med Genomics 2016; 9(1): 65. http://dx.doi.org/10.1186/s12920-016-0225-2 PMID: 27756306

[5] Mármol I, Sánchez-de-Diego C, Pradilla Dieste A, Cerrada E, Rodriguez Yoldi MJ. Colorectal carcinoma: a general overview and future perspectives in colorectal cancer. Int J Mol Sci 2017; 18(1): 197. http://dx.doi.org/10.3390/ijms18010197 PMID: 28106826

[6] Granados-Romero JJ, Valderrama-Treviño AI, Contreras-Flores EH, *et al.* Colorectal cancer: a review. Int J Res Med Sci 2017; 5(11): 4667-76. http://dx.doi.org/10.18203/2320-6012.ijrms20174914

[7] WHO, WHO: Cancer. Available at https://www.who.int/news-room/fact-sheets/detail/cancer. [Cited August 14, 2019].

[8] DeSantis CE, Miller KD, Goding Sauer A, Jemal A, Siegel RL. Cancer statistics for African Americans, 2019. CA Cancer J Clin 2019; 69(3): 211-33. http://dx.doi.org/10.3322/caac.21555 PMID: 30762872

[9] American Cancer Society. Cancer facts & figures 2015. American Cancer Society 2015.

[10] May FP, Anandasabapathy S. Colon cancer in Africa: Primetime for screening? Gastrointest Endosc 2019; 89(6): 1238-40. http://dx.doi.org/10.1016/j.gie.2019.04.206 PMID: 31104752

[11] World Health Organization. National cancer control programmes: policies and managerial guidelines. World Health Organization 2002.

[12] Golub TR, Slonim DK, Tamayo P, *et al.* Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 1999; 15286(5439): 531-7.

[13] Wang J, Tan AC, Tian T, Eds. Next generation microarray bioinformatics: methods and protocols. Humana Press 2012. http://dx.doi.org/10.1007/978-1-61779-400-1

[14] Tan AC, Gilbert D. Ensemble machine learning on gene expression data for cancer classification. Appl Bioinformatics 2003; 2(3 Suppl): S75-83.

[15] Ca DA, Mc V. Gene expression data classification using support vector machine and mutual information-based gene selection. Procedia Comput Sci 2015; 47: 13-21. http://dx.doi.org/10.1016/j.procs.2015.03.178

[16] Blagus R, Lusa L. Class prediction for high-dimensional class-imbalanced data. BMC Bioinformatics 2010; 11(1): 523. http://dx.doi.org/10.1186/1471-2105-11-523 PMID: 20961420

[17] Rajeswari P, Reena GS. Human liver cancer classification using microarray gene expression data. Int J Comput Appl 2011; 34(6): 25-37.

[18] Datta S. Statistical analysis of next generation sequencing data. New York: Springer 2014. http://dx.doi.org/10.1007/978-3-319-07212-8

[19] Rai MF, Tycksen ED, Sandell LJ, Brophy RH. Advantages of RNA-seq compared to RNA microarrays for transcriptome profiling of anterior cruciate ligament tears. J Orthop Res 2018; 36(1): 484-97. PMID: 28749036

[20] Castillo D, Galvez JM, Herrera LJ, *et al.* Leukemia multiclass assessment and classification from microarray and RNA-seq technologies integration at gene expression level. PLoS One 2019; 14(2): e 0212127. http://dx.doi.org/10.1371/journal.pone.0212127 PMID: 30753220

[21] Zhang W, Yu Y, Hertwig F, *et al.* Comparison of RNA-seq and microarray-based models for clinical endpoint prediction. Genome Biol 2015; 16(1): 133. http://dx.doi.org/10.1186/s13059-015-0694-1 PMID: 26109056

[22] Dong K, Zhao H, Tong T, Wan X. NBLDA: negative binomial linear discriminant analysis for RNA-Seq data. BMC Bioinformatics 2016; 17(1): 369. http://dx.doi.org/10.1186/s12859-016-1208-1 PMID: 27623864

[23] Zararsız G, Goksuluk D, Korkmaz S, *et al.* A comprehensive simulation study on classification of RNA-Seq data. PLoS One 2017; 12(8): e0182507. http://dx.doi.org/10.1371/journal.pone.0182507 PMID: 28832679

[24] Kulski J, Ed. Next Generation Sequencing: Advances, Applications and Challenges. BoD–Books on Demand 2016.14. http://dx.doi.org/10.5772/60489

[25] Wang S, Huang H, Han R, *et al.* BpAP1 directly regulates BpDEF to promote male inflorescence formation in Betula platyphylla × B. pendula. Tree Physiol 2019; 39(6): 1046-60. http://dx.doi.org/10.1093/treephys/tpz021 PMID: 30976801

[26] Aziz R, Verma CK, Srivastava N. Artificial neural network classification of high dimensional data with novel optimization approach of dimension reduction. Annals of Data Science 2018; 5(4): 615-35. http://dx.doi.org/10.1007/s40745-018-0155-2

[27] Salem H, Attiya G, El-Fishawy N. Classification of human cancer diseases by gene expression profiles. Appl Soft Comput 2017; 50: 124-34. http://dx.doi.org/10.1016/j.asoc.2016.11.026

[28] Jain I, Jain VK, Jain R. Correlation feature selection based improved-binary particle swarm optimization for gene selection and cancer classification. Appl Soft Comput 2018; 62: 203-15. http://dx.doi.org/10.1016/j.asoc.2017.09.038

[29] Anders S, Huber W. Differential expression analysis for sequence count data. Nat Prec 2010. https://doi.org/10.1038/npre.2010.4282.2

[30] Taminau J, Lazar C, Meganck S, Nowé A. Comparison of merging and meta-analysis as alternative approaches for integrative gene expression analysis. ISRN Bioinform 2014; 2014: 345106. http://dx.doi.org/10.1155/2014/345106 PMID: 25937953

[31] Lai Y. Differential expression analysis of digital gene expression data: RNA-tag filtering, comparison of t-type tests and their genome-wide co-expression based adjustments. Int J Bioinform Res Appl 2010; 6(4): 353-65. http://dx.doi.org/10.1504/IJBRA.2010.035999 PMID: 20940123

[32] Castillo D, Gálvez JM, Herrera LJ, Román BS, Rojas F, Rojas I. Integration of RNA-Seq data with heterogeneous microarray data for breast cancer profiling. BMC Bioinformatics 2017; 18(1): 506. http://dx.doi.org/10.1186/s12859-017-1925-0 PMID: 29157215

[33] Gomez-Cabrero D, Abugessaisa I, Maier D, *et al.* Data integration in the era of omics: current and future challenges. BMC Syst Biol 2014; 8(Suppl 2). http://dx.doi.org/10.1186/1752-0509-8-S2-I1

[34] Huang DW, Sherman BT, Tan Q, *et al.* DAVID bioinformatics resources: expanded annotation database and novel algorithms to better extract biology from large gene lists. Nucleic Acids Res 2007; 135(Suppl_2): W169-W175. http://dx.doi.org/10.1093/nar/gkm415

[35] Bolstad BM, Irizarry RA, Åstrand M, Speed TP. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. Bioinformatics 2003; 19(2): 185-93. http://dx.doi.org/10.1093/bioinformatics/19.2.185 PMID: 12538238

[36] Witten DM. Classification and clustering of sequencing data using a Poisson model. Ann Appl Stat 2011; 5(4): 2493-518. http://dx.doi.org/10.1214/11-AOAS493

[37] Boser BE, Guyon IM, Vapnik VN. A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning theory. 1992; pp. 144-52. http://dx.doi.org/10.1145/130385.130401

[38] Moguerza JM, Muñoz A. Support vector machines with applications. Stat Sci 2006; 21(3): 322-36. http://dx.doi.org/10.1214/088342306000000493

[39] Stephens D, Diesing M. A comparison of supervised classification methods for the prediction of substrate type using multibeam acoustic and legacy grain-size data. PLoS One 2014; 9(4): e93950. http://dx.doi.org/10.1371/journal.pone.0093950 PMID: 24699553

[40] Brown MP, Grundy WN, Lin D, *et al.* Support vector machine classification of microarray gene expression data. University of California, Santa Cruz, Technical Report UCSC-CRL-99-09 1999 Jun; 12.

[41] Chu F, Wang L. Gene expression data analysis using support vector machines. Proceedings of the International Joint Conference on Neural Networks, 2003.

[42] Karatzoglou A, Smola A, Hornik K, Karatzoglou MA. Package 'kernlab'. Technical report, CRAN, 03 2016 2019 Nov; 12.

[43] Friedman J, Hastie T, Tibshirani R. The elements of statistical learning. Springer series in statistics. New York 2001.

[44] Breiman L. Random forests. Mach Learn 2001; 45(1): 5-32. http://dx.doi.org/10.1023/A:1010933404324

[45] Qi Y. Random forest for bioinformatics in ensemble machine learning. Boston, MA: Springer 2012; pp. 307-23.

[46] Chen X, Ishwaran H. Random forests for genomic data analysis. Genomics 2012; 99(6): 323-9. http://dx.doi.org/10.1016/j.ygeno.2012.04.003 PMID: 22546560

[47] Pappu V, Pardalos PM. High-dimensional data classification InClusters, Orders, and Trees: Methods and Applications. New York, NY: Springer 2014; pp. 119-50.

[48] Do TN, Lenca P, Lallich S, Pham NK. Classifying very-high-dimensional data with random forests of oblique decision trees InAdvances in knowledge discovery and management. Berlin, Heidelberg: Springer 2010; pp. 39-55.

[49] RColorBrewer S, Liaw MA. Package 'randomForest'. Berkeley, CA, USA.: University of California Berkeley 2018.22.

[50] Dwivedi AK. Artificial neural network model for effective cancer classification using microarray gene expression data. Neural Comput Appl 2018; 29(12): 1545-54. http://dx.doi.org/10.1007/s00521-016-2701-1

[51] Ripley B, Venables W, Ripley MB. Package 'nnet'. R package version 2016 Feb; 27: 3-12.

[52] De Campos LM, Cano A, Castellano JG, Moral S. Bayesian networks classifiers for gene-expression data. 11th International Conference on Intelligent Systems Design and Applications. IEEE 2011. http://dx.doi.org/10.1109/ISDA.2011.6121822

[53] Yao Z, Ruzzo WL. A regression-based K nearest neighbor algorithm for gene function prediction from heterogeneous data. BMC Bioinformatics 2006; 7: 11. http://dx.doi.org/10.1186/1471-2105-7-S1-S11

[54] Tharwat A. Classification assessment methods. Appl Comput Informatics 2018; 17(1): 168-92. http://dx.doi.org/10.1016/j.aci.2018.08.003

[55] Mohammed M, Mwambi H, Omolo B, Elbashir MK. Using stacking ensemble for microarray-based cancer classification. International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE). IEEE 2018; 1-8. http://dx.doi.org/10.1109/ICCCEEE.2018.8515872

[56] Li W. Volcano plots in analyzing differential expressions with mRNA microarrays. J Bioinform Comput Biol 2012; 10(6): 1231003. http://dx.doi.org/10.1142/S0219720012310038 PMID: 23075208

[57] Landau WM, Liu P. Dispersion estimation and its effect on test performance in RNA-seq data analysis: a simulation-based comparison of methods. PLoS One 2013; 8(12): e81415. http://dx.doi.org/10.1371/journal.pone.0081415 PMID: 24349066

[58] Dry JR, Pavey S, Pratilas CA, *et al.* Transcriptional pathway signatures predict MEK addiction and response to selumetinib (AZD6244). Cancer Res 2010; 70(6): 2264-73. http://dx.doi.org/10.1158/0008-5472.CAN-09-1577 PMID: 20215513

[59] Lundberg SM, Lee SI. A unified approach to interpreting model predictions. InAdvances in neural information processing systems. 2017; pp. 4765-74.

[60] Statnikov A, Henaff M, Narendra V, *et al.* A comprehensive evaluation of multicategory classification methods for microbiomic data. Microbiome 2013; 1(1): 11. http://dx.doi.org/10.1186/2049-2618-1-11 PMID: 24456583

[61] Mamatjan Y, Agnihotri S, Goldenberg A, *et al.* Molecular signatures for tumor classification: an analysis of the cancer genome atlas data. J Mol Diagn 2017; 19(6): 881-91. http://dx.doi.org/10.1016/j.jmoldx.2017.07.008 PMID: 28867603

[62] Rashid M, Vishwakarma RK, Deeb AM, Hussein MA, Aziz MA. Molecular classification of colorectal cancer using the gene expression profile of tumor samples. Exp Biol Med (Maywood) 2019; 244(12): 1005-16. http://dx.doi.org/10.1177/1535370219850788 PMID: 31091989

[63] Popovici V, Budinska E, Tejpar S, *et al.* Identification of a poor-prognosis BRAF-mutant-like population of patients with colon cancer. J Clin Oncol 2012; 30(12): 1288-95. http://dx.doi.org/10.1200/JCO.2011.39.5814 PMID: 22393095

[64] Zumwalt TJ, Shigeyasu K, Weng W, Okugawa Y, Miyoshi J, Goel A. The ATP6V1C2 (a vacuolar-ATPase gene) is a novel early prognosticator for colorectal cancer. Cancer Res 2016. DOI: 10.1158/1538-7445.AM2016-768

[65] He WL, Weng XT, Wang JL, *et al.* Association between c-Myc and colorectal cancer prognosis: a meta-analysis. Front Physiol 2018; 9: 1549. http://dx.doi.org/10.3389/fphys.2018.01549 PMID: 30483143

[66] Liu H, Gu X, Wang G, *et al.* Copy number variations primed lncRNAs deregulation contribute to poor prognosis in colorectal cancer. Aging (Albany NY) 2019; 11(16): 6089-108. http://dx.doi.org/10.18632/aging.102168 PMID: 31442207

[67] Zhang TM, Huang T, Wang RF. Cross talk of chromosome instability, CpG island methylator phenotype and mismatch repair in colorectal cancer. Oncol Lett 2018; 16(2): 1736-46. http://dx.doi.org/10.3892/ol.2018.8860 PMID: 30008861

[68] Zheng H, Ying H, Wiedemeyer R, *et al.* PLAGL2 regulates Wnt signaling to impede differentiation in neural stem cells and gliomas. Cancer Cell 2010; 17(5): 497-509. http://dx.doi.org/10.1016/j.ccr.2010.03.020 PMID: 20478531

[69] Su C, Li D, Li N, *et al.* Studying the mechanism of PLAGL2 overexpression and its carcinogenic characteristics based on 3′-untranslated region in colorectal cancer. Int J Oncol 2018; 52(5): 1479-90. http://dx.doi.org/10.3892/ijo.2018.4305 PMID: 29512763

scientific reports

Check for updates

OPEN

A stacking ensemble deep learning approach to cancer type classification based on TCGA data

Mohanad Mohammed[1]✉, Henry Mwambi[1], Innocent B. Mboya[1,4], Murtada K. Elbashir[5,6] & Bernard Omolo[1,2,3]

Cancer tumor classification based on morphological characteristics alone has been shown to have serious limitations. Breast, lung, colorectal, thyroid, and ovarian are the most commonly diagnosed cancers among women. Precise classification of cancers into their types is considered a vital problem for cancer diagnosis and therapy. In this paper, we proposed a stacking ensemble deep learning model based on one-dimensional convolutional neural network (1D-CNN) to perform a multi-class classification on the five common cancers among women based on RNASeq data. The RNASeq gene expression data was downloaded from Pan-Cancer Atlas using *GDCquery* function of the *TCGAbiolinks* package in the *R* software. We used least absolute shrinkage and selection operator (LASSO) as feature selection method. We compared the results of the new proposed model with and without LASSO with the results of the single 1D-CNN and machine learning methods which include support vector machines with radial basis function, linear, and polynomial kernels; artificial neural networks; k-nearest neighbors; bagging trees. The results show that the proposed model with and without LASSO has a better performance compared to other classifiers. Also, the results show that the machine learning methods (SVM-R, SVM-L, SVM-P, ANN, KNN, and bagging trees) with under-sampling have better performance than with over-sampling techniques. This is supported by the statistical significance test of accuracy where the *p*-values for differences between the SVM-R and SVM-P, SVM-R and ANN, SVM-R and KNN are found to be $p = 0.003$, $p = < 0.001$, and $p = < 0.001$, respectively. Also, SVM-L had a significant difference compared to ANN $p = 0.009$. Moreover, SVM-P and ANN, SVM-P and KNN are found to be significantly different with *p*-values $p = < 0.001$ and $p = < 0.001$, respectively. In addition, ANN and bagging trees, ANN and KNN were found to be significantly different with *p*-values $p = < 0.001$ and $p = 0.004$, respectively. Thus, the proposed model can help in the early detection and diagnosis of cancer in women, and hence aid in designing early treatment strategies to improve survival.

Recent global public health research shows an epidemiological paradigm shift from infectious to non-communicable diseases, the latter including different types of cancers. The incidence and prevalence of cancer are on the increase worldwide, both in the developing and developed countries[1,2]. The global cancer statistics estimated about 19.3 million new cancer cases in 2020 alone, and close to 10 million deaths of 36 cancers in 185 countries[3]. Breast cancer (with estimated 2.3 million new cases) is the most common diagnosed among women, followed by lung, colorectal, thyroid, and ovarian cancers. Moreover, the most leading cause of death is the lung cancer (with estimated 1.8 million deaths). The cancer burden is expected to increase to 28.4 million cases by 2040[3].

Cancer tumor classification based on morphological characteristics alone has serious limitations in differentiating among cancer tumors and may cause a strong bias in identifying the tumor by experts[4–6]. Recently, RNASeq gene expression data[7,8] has emerged as the preferred technology for the simultaneous quantification of gene expression compared to the DNA microarray[9,10]. The classification of cancer using gene expression data

[1]School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Pietermaritzburg, Private Bag X01, Scottsville 3209, South Africa. [2]Division of Mathematics and Computer Science, University of South Carolina-Upstate, 800 University Way, Spartanburg, USA. [3]School of Public Health, Faculty of Health Sciences, University of Witwatersrand, Johannesburg, South Africa. [4]Department of Epidemiology and Biostatistics, Kilimanjaro Christian Medical University College (KCMUCo), P. O. Box 2240, Moshi, Tanzania. [5]College of Computer and Information Sciences, Jouf University, Sakaka 72441, Saudi Arabia. [6]Faculty of Mathematical and Computer Sciences, University of Gezira, Wad Madani 11123, Sudan. ✉email: mohanadadam32@gmail.com

from RNASeq technology provides the opportunity to discriminate healthy and diseased samples or among different types and subtypes of cancer more accurately[11]. RNASeq gene expression data have had a profound impact on disease diagnoses and prognoses through accurate disease classification, which has helped clinicians to choose the appropriate treatment plans for patients[12]. There exists striking disparities in the global cancers among women[3,13]. Correct classification of these cancers is among the essential strategies to inform clinical decisions and reduce morbidity and mortality from cancers among women.

Although the use of gene expression data from RNASeq technology has improved cancer classification, it has its own limitations due to it being characterized by small samples sizes, with each sample having a large number of genes (the curse of dimensionality)[14,15]. In addition, the samples also contain several genes that are uninformative and degrade the classification performance[11,16]. As a way to mitigate this problem, it has been suggested to first perform filtration and feature selection through methods such as the two-sample *t*-test at a given stringent significance threshold before going further with model building[17]. This procedure ensures that only informative and sufficiently differentially expressed genes between the outcome classes are used in building the classifiers. This process of feature selection motivates the evaluation of methods for the classification of different cancer tumors and disease stages, to improve early detection and the design of targeted treatment strategies that may reduce mortality. The two-sample *t*-test as a method for feature selection is easy to use but comes with the problem of multiple testing that the user has to deal with. Other methods or approaches that are model based, such as regularized regression methods, have recently become popularly used.

There are many supervised and unsupervised machine learning as well as deep learning methods developed for cancer classification using gene expression data. Several studies reported a higher predictive performance of the machine learning methods on the multi-class cancer classification problem[11,18–20]. These studies, however, differ in the methods used for feature (gene) selection. In particular, Castillo et al.[18] used differential expression analysis and minimum-redundancy maximum-relevance method for feature selection in the microarray and RNASeq data. García-Díaz et al.[11] applied a grouping genetic algorithm for feature selection in five different cancers using RNASeq data.

Ramaswamy et al.[19], on the other hand, used support vector machines (SVM) and a recursive feature elimination method to remove the uninformative genes. These studies concentrated on the application of machine learning methods on a multi-class classification problem. Several methods developed by other authors for multi-class cancer classification are reported to have a higher predictive performance compared to existing methods[21]. Lee et al.[22] proposed a new ensemble classifier called cancer predictor using an ensemble model (CPEM), for classification of over 31 different cancer tumors downloaded from TCGA repository. In addition, they assessed different input features such as mutation profiles, mutations rates, mutation spectra, and signature. Thereafter, they investigated different machine learning and feature selection models in order to find the best model which achieved 84% of accuracy using 10 folds cross-validation. Furthermore, they used the six most common cancers out of 31 types and the model achieved 94% of classification accuracy. However, some of the statistical methods achieved results that are better than machine learning algorithms.

Tabares-Soto et al.[23] compared machine learning and deep learning methods in classifying 11 different tumor classes using microarray gene expression data. They implemented eight supervised machine learning methods including KNN, support vector classifier (SVC), logistics regression (LR), linear discriminant analysis (LDA), naïve Bayesian classifier (NB), multi-layer perceptron (MLP), decision trees, and random forest (RF) as well as one unsupervised method such as k-means. In addition, they applied two deep neuronal networks (DNN) methods. Their results showed that the deep learning methods outperformed the other machine learning methods.

In this study, we propose a stacking ensemble deep learning model that uses five 1D-CNN as base models. The results of these models are combined using NN, which is used as a meta model to classify the most common types of cancers among women using RNASeq data. We compared the performance of our new proposed model when using the full list of genes as input with its performance when using a reduced selection of genes using LASSO. Also, we consider comparing the performance of our current proposed model with other machine learning methods since there are limited studies that compare the performance of deep learning and machine learning methods to classify different types of cancer. LASSO is used as a feature selection technique, since it has been shown to improve prediction accuracy, especially when there is a small number of observations and a large number of features[24]. Findings from this study might help in the early detection and accurate classification of these cancer types and contribute to efforts of finding therapies that may increase survival for women at risk.

Material and methods

In this paper, we downloaded the RNASeq gene expression data from Pan-Cancer Atlas (https://portal.gdc.cancer.gov/), using *R* statistical software version 3.6.3 via the *TCGAbiolinks* package[25–27]. The data contains 2166 samples from the top five common cancers between women. We applied eight multi-class classification methods to find the best classifier that discriminates among five common cancers among women. The machine learning methods were implemented in the *R* software, while the deep learning method (1D-CNN) was implemented using *TensorFlow* with *Keras*.

Datasets. We used only five cancer tumors (normal cases were excluded) from RNASeq gene expression datasets. The cancer tumors were breast, colon adenocarcinoma, ovarian, lung adenocarcinoma, and thyroid cancer. The datasets were downloaded from Pan-Cancer Atlas using *GDCquery* function of the *TCGAbiolinks* package in *R*[26]. *GDCquery* function has many parameters, to define the function known by the following names: project, legacy, data.category, data.type, platform, file.type, experimental.strategy, and sample.type. The project parameter indicates a list of the data that should be downloaded. In our case, we passed the five project codes corresponding to our five types of cancer, which are TCGA-BRCA, TCGA-COAD, TCGA-OV, TCGA-LUAD,

Cancer tumor	Number of samples (%)	Training (≈ 70%)	Testing (≈ 30%)
Breast (BRCA)	1082 (50)	753	329
Colon adenocarcinoma (COAD)	135 (6)	99	36
Lung adenocarcinoma (LUAD)	275 (13)	189	86
Ovarian (OV)	304 (14)	217	87
Thyroid (THCA)	370 (17)	259	111
Total	2166	1517	649

Table 1. Number of samples in each class used in the classification.

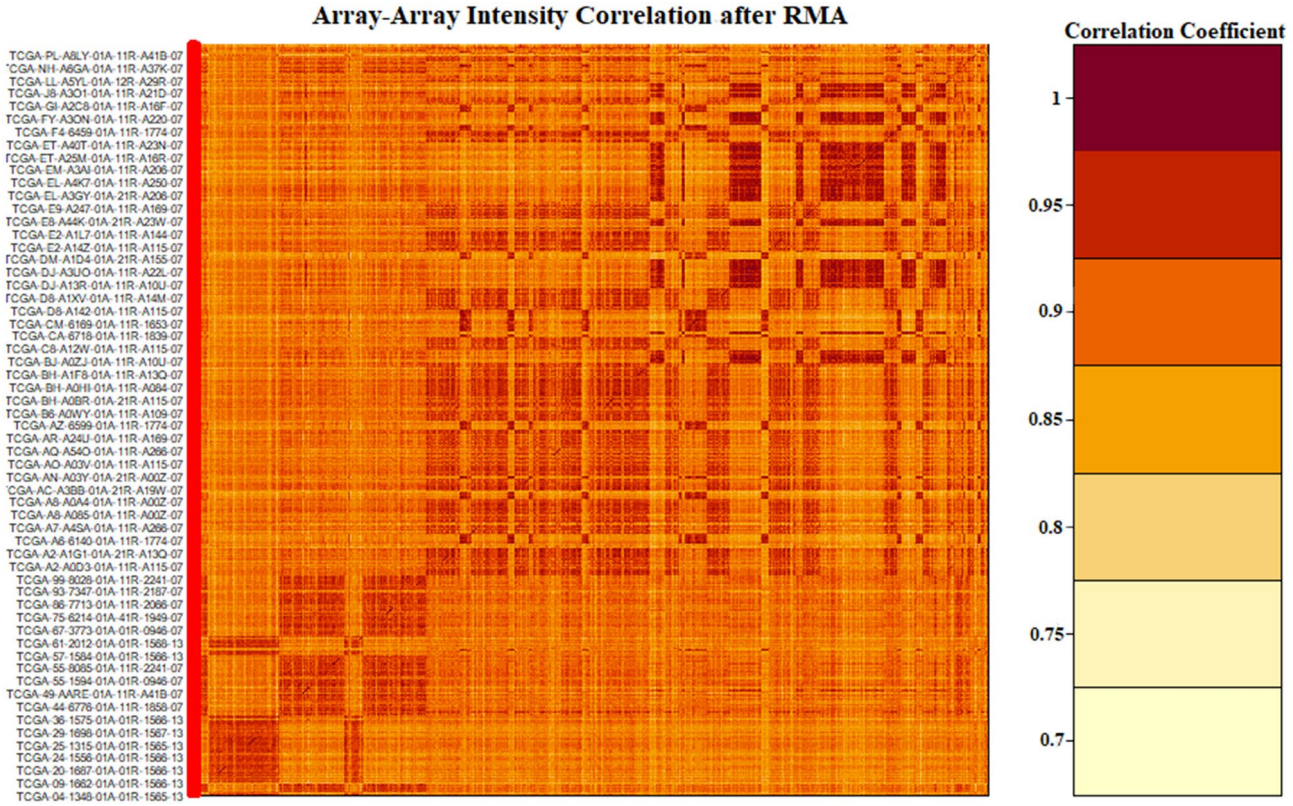


Figure 1. Array-array intensity correlation (AAIC) matrix defines the Pearson correlation coefficients among the samples.

and TCGA-THCA. We set the legacy to "true", which helps the query to search only in the legacy repository for the unmodified stored data in the TCGA data portal.

"Gene expression" and "Gene expression quantification" are passed to the data.category and data.type arguments, respectively, to filter the data files to be downloaded. The platform "Illumina HiSeq" was used to download the data. We used "results" for file.type argument to filter the legacy database, and "RNA-Seq" was chosen as experimental.strategy argument to produce the expression profiles. Moreover, we selected the tumor samples to be downloaded using the "Primary solid Tumor" value as sample.type argument. The downloaded data in a matrix form included five types of cancer, where the columns represent the samples and the rows containing the genes, i.e. features (equivalently covariates). The datasets were combined to give 2166 tumor samples obtained from all the five cancers, with 19,947 common genes. Due to the curse of high dimensionality, we performed filtration and feature selection to reduce the high number of genes in order to exclude irrelevant and noisy ones that could affect the performance of the methods. Thus, we applied normalization, transformation, and filtration steps to the data to select the informative genes that potentially could contribute positively to the classification accuracy. Table 1 below shows a summary of the downloaded data including the training and testing fractions for each cancer tumor.

Data pre-processing. We used *TCGAanalyze_Preprocessing* function in *TCGAbiolinks* package[26], which utilizes an array-array intensity correlation (AAIC) approach to obtain a $N \times N$ square symmetric matrix of Spearman correlations among the samples. The AAIC enabled us to find samples with low correlation considered as possible outliers (Fig. 1). After that, we performed gene normalization through TCGAanalyze_Normalization

function, which calls the sub-routines newSeqExpressionSet, withinLaneNormalization, betweenLaneNormalization, and counts from *EDASeq* package to adjust the GC-content effect or other gene level effects, distributional differences between lanes, and global-scaling and full-quantile normalization[28]. *TCGAanalyze_Filtering* was used for filtering out the irrelevant genes and returned the genes with the mean intensity across the samples higher than 0.25, which was the threshold defined quantile mean. After applying this process, we found 14,899 genes to be informative meaning 5048 genes were rendered irrelevant. For further reduction and precise differential gene expression analysis, we used *DESeq* package in *R*[29–31]. *DESeq* analyses the gene expression based on the negative binomial distribution and a shrinkage estimator for the distribution's variance. After using *DESeq* package, 12,649 genes out of the 14,899 post initial filtering were found to be differentially expressed meaning a further 2250 genes were removed.

Feature selection using LASSO regression. The RNASeq gene expression data after preprocessing had 12,649 dimensions or features, which was still huge given that the number of samples was 2166. Therefore, LASSO regression was used to decrease the number of genes or features that enabled us to effectively analyze the data. LASSO is a method that performs regularization and feature selection through a shrinkage (regularization) process. LASSO penalizes the regression coefficients with $L1$-norm whereby some coefficients are shrunk to zero. After that, the coefficients of the regression variables having significantly non-zero values are selected and used in the model[24].

In the case of the multinomial response with $K > 2$ levels, assume that $p_\ell(g_i) = \Pr(C = c_i|g_i)$, where $c_i \in \{1, 2, 3, \ldots, K\}$ is the ith response. The log-likelihood of the multinomial model under LASSO model can be written in a generalized form as[32]

$$\underset{\{\beta_{0\ell}, \beta_\ell\}ss_1^K \in \mathbb{R}^{K(p+1)}}{max} \left[\frac{1}{N}\sum_{i=1}^{N} \log p_{c_i}(g_i) - \lambda \sum_{\ell=1}^{K} P_\alpha(\beta_\ell)\right]. \tag{1}$$

which can be maximized as a penalized log-likelihood.

The outcomes in the data can be denoted in the form of a matrix Y of dimension $N \times K$, with elements $y_{i\ell} = I(c_i = \ell)$. Thus, the terms in the regularized log-likelihood in Eq. (1) can be written in more explicit form

$$\ell(\{\beta_{0\ell}, \beta_\ell\}_1^K) = \frac{1}{N}\sum_{i=1}^{N}\left[\sum_{\ell=1}^{K} y_{i\ell}\left(\beta_{0\ell} + g_i^T\beta_\ell\right) - \log\left(\sum_{\ell=1}^{K} e^{\beta_{0\ell}+g_i^T\beta_\ell}\right)\right]. \tag{2}$$

$$P_\alpha(\beta) = (1-\alpha)\frac{1}{2}\|\beta\|_{L_2}^2 + \alpha\|\beta\|_{L_1}, \tag{3}$$

$$= \sum_{j=1}^{p}\left[\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j|\right]. \tag{4}$$

P_α is the penalty part, where g_i is the gene expression levels for sample i, β_ℓ is the vector of the regression coefficients, $y_{i\ell}$ is the class response value in sample i. When $\alpha = 0$ in Eq. (3) we obtain the ridge regression penalty, whereas $\alpha = 1$ leads to LASSO regression penalty.

We chose LASSO regression because it uses the sum of the absolute values of the model parameters, restricted to be less than a fixed value as the penalty. LASSO, with tenfold cross-validation returned 173 genes (Supplementary File 1). These genes were obtained when lambda (λ) value gave a minimal deviance associated with the response variable, and so were used for the classification. The cross-validated multinomial deviance is a function of log(λ), and when log(λ) is equal to -1, it is an indication that λ and multinomial deviance are both big. As λ decreases and becomes very small, the multinomial deviance also becomes small and almost flat, indicating that the attained model is a good fit.

There are many advantages of the LASSO method, which include removing those variables with zero coefficients that lead to reduced variance without an intrinsic increase in bias. The method also minimizes over-fitting by excluding irrelevant variables that are not related to the outcome variable. The LASSO method naturally also deals with the multiple testing problem, by penalizing irrelevant features, whose contribution is shrunk to zero. This leads to an improved classification and prediction accuracy[24,33]. In our case, LASSO was implemented using *glmnet package* in *R*[34].

Data partitioning. We used tenfold cross-validation to evaluate the different prediction methods using 70% of the dataset. In the tenfold cross-validation, the dataset is divided into ten parts, where one part is removed to represent the validation set, and the remaining nine parts combined to represent the training set. Thus, this process is repeated ten times by removing one part each time to have a different part of the data for validation[35]. We left aside 30% of the entire dataset, which served as an independent testing set for the final evaluation.

The classification models. We performed classification on the different cancers as a multi-classification problem using gene expression levels as covariates. Eight classification methods were used: the new proposed stacking ensemble deep learning model; one-dimensional convolutional neural network (1D-CNN); support vector machines (SVM) with radial basis function, linear, and polynomial kernels; artificial neural networks (ANN); K-nearest neighbors (kNN); and bagging trees.

Support vector machines (SVM)[36], is a well-known machine learning method that has been used widely in many fields, including gene expression data analysis[37,38]. SVM aims to find an optimal hyperplane that separates the data into two different classes for the binary classification problem, determined by a subset of samples known as support vectors[39]. SVM can handle non-linearly separable problems by transforming the data using mapping kernel functions. These functions include radial basis, polynomial, and linear functions[40]. The SVM is implemented using *kernlab* package in *R* statistical software[41].

Suppose we have n samples and p genes. Furthermore, assume samples belong to two linearly separable classes represented by + 1 or − 1, and suppose $\boldsymbol{g}_i$ is the features vector. Then we let, $(\boldsymbol{g}_i, \boldsymbol{y}_i) \in G \times Y, i = 1, 2, 3, \ldots, n$, where $y_i \in \{+1, -1\}$ is the target variable dichotomy in the p dimensional space. The aim is to classify the sample into one of the two classes and by extension find an SVM classifier that generalizes to a multi-class problem. There are many hyperplanes that discriminate the two classes, but the goal is achieved by finding an optimal separating hyperplane that lies furthest from the both classes.

The separating hyperplane can be defined by

$$\boldsymbol{w} * \boldsymbol{g} + b = 0. \tag{5}$$

where $\boldsymbol{w}$ is the weight vector, b is the bias, and $|b|/\|\boldsymbol{w}\|$ is the perpendicular distance to the hyperplane. We can rescale the $\boldsymbol{w}$ and b such that the following equation determines the point in each class that is nearest to the hyperplane defined by the equation

$$\left|\boldsymbol{w} * \boldsymbol{g} + b\right| = 1. \tag{6}$$

Therefore, a separating hyperplane for the two classes should follow

$$\boldsymbol{w} * \boldsymbol{g} + b \geq +1, \quad when\, y_i = +1. \tag{7}$$

$$\boldsymbol{w} * \boldsymbol{g} + b \leq -1, \quad when\, y_i = -1. \tag{8}$$

After the rescaling, the distance from the nearest point in each class to the hyperplane becomes $1/\|\boldsymbol{w}\|$. Consequently, the distance between the two classes is $2/\|\boldsymbol{w}\|$, which is called the margin. The solution of the following optimization problem is obtained by maximizing the margin:

$$\begin{aligned} &\min_{w,b} \|\boldsymbol{w}\|^2 \\ &\text{subject to} \quad y_i\left(w * g + b\right) \geq 1, \quad i = 1, 2, 3, \ldots, n. \end{aligned} \tag{9}$$

For the multi-class problem there are many types of extensions that can be used such as one-vs-one, one-vs-all (one-vs-rest), decision directed acyclic graph based approach, multi-class objective function, and error-correcting output code based approach. These approaches use the same binary classification principle, where the multi-class problem is decomposed into multiple binary problems. In the one-vs-one multi-class classification problem the SVM classifier produces all possible pairs of binary classifications. Suppose we have k classes where $k > 2$, then, $\frac{k(k-1)}{2}$ binary classifiers are produced in the training step of the algorithm. Consequently, a sample in the test dataset is assigned the class label that is voted the most by the $\binom{k}{2}$ binary classifiers from the trained one-vs-one SVM. In our case we use the one-vs-one multi-class classifier.

Artificial neural networks (ANN) is a computational method constructed from many layers, each layer consisting of nodes called neurons[42]. The data flows from the input layer to the output layer through the hidden layers[43]. The nodes between the input through the hidden layers to the output layers are connected by appropriately defined weights or weight functions. The number of input and output layers depends on the number of covariates in the dataset as well as a number of classes in the outcome variable[43]. The inputs are weighted by multiplying every one of them by a weight which is a measure of its contribution. Therefore, the hidden layer receives the weighted inputs and produce outputs using an activation function(s)[40,42]. ANN can be implemented using the *R* package *nnet*[44].

Specifically suppose we have gene expression data with p genes. The input layer receives the p genes and multiplies them by weights as follow

$$b_i = \sum_{j=0}^{p} w_{ij}^{(1)} g_i \quad i = 1, 2, 3, \ldots, n, \tag{10}$$

where g is a vector of input features and $g_0 = 1$ is a constant input feature with weight w_{i0}. The b_i are called activations, and the parameters $w_{ij}^{(1)}$ are the weights. The subscripts (1) refer to the first layer of the network. Then the activations are transformed by a nonlinear activation function f, usually a sigmoid function as given in the following equation

$$z_i = f(b_i) = \frac{1}{1 + \exp(-b_i)}. \tag{11}$$

In the second layer, the outputs of the hidden units are linearly combined to give the activations

$$h_k = \sum_{i=0}^{n} w_{ik}^{(2)} z_i \quad k = 1, 2, 3, \ldots, K, \tag{12}$$

where the $w_{ik}^{(2)}$ are the weight parameters for the transformation in the second layer of the neural network. The outputs are transformed using an activation function such as the sigmoid function

$$y_k = f(h_k) = \frac{1}{1 + \exp(-h_k)}. \tag{13}$$

K-nearest neighbors (kNN) is a non-parametric method used for classification and regression[45]. The idea behind kNN lies in finding the most nearest neighbors of the new sample, and this is based on the similarity and distance metric[46]. In kNN, *k*-neighbors determine the class of a new instance; therefore, the new sample is assigned the class that is most likely among the *k*-neighbors[40,42]. In general, kNN has two phases; the first is finding the nearest neighbors, and the second is assigning the class of a new sample using those neighbors by the majority vote rule. kNN is implemented using *R* package *caret*[47].

Suppose we have two samples s_1, s_2 each with p genes. Since kNN uses the Euclidean distance measure to find the closest sample for a new sample, the distance between the two samples can be calculated as

$$dist(s_1, s_2) = \sqrt{\sum_{j=1}^{p} \left(g_{1j} - g_{2j}\right)^2}. \tag{14}$$

A new sample is allocated the class that most of its neighbors fall, that is, model class of its neighbors.

Bagging trees or bootstrap aggregation method is appealing because its ability to reduce the variance associated with a prediction and hence, improve the prediction accuracy[48]. The method splits the data into many bootstrap samples, thereafter, train the model for each bootstrap. Then, the overall prediction obtained by averaging and voting for regression and classification, respectively.

Convolution Neural Networks (CNNs) are deep learning architectures that have multi-layers between the input and output and are designed for image analysis and classification[49–51]. Deep learning is applied successfully in many areas including medical image analysis, computer vision, drug design, and bioinformatics and yield performance that sometimes surpass expert personals' performance[52]. CNNs are a regularized version of fully connected networks (multilayer perceptrons), in which each neuron in one layer is connected to all the neurons in the layer that follows it. The connectivity between the neurons is inspired by the biological process and resembles the arrangement of the animal visual cortex. In contrast to other image classification and analysis algorithms, CNNs use little pre-processing by learning the filters that capture temporal and special dependencies in an image instead of hand-engineering them. A sequence of stacked layers (convolutional layer, pooling layer, and fully-connected layer) makes the architecture of CNNs and in each layer, a differentiable function is used to transform one volume of activations to the layer that follows it. The major building blocks in CNNs are the convolutional layers, which apply filters on an input image to create a feature map. To get a good classification performance, CNNs normally decrease the features of the image into an easier processed arrangement without dropping essential features. The pooling layers use max pooling or average pooling to reduce the dimension of the image's features. The fully connected layer is an important component in the CNNs architecture that derives the final classification results.

The input to the CNNs is a tensor of order 3 that represents an image having m rows and n columns with 3 color channels (RGB). The tensor encodes the pixel intensities of the image and produces the input features that go through the convolutional, pooling, and the fully connected layers sequentially. In the convolutional layer, a filter of size f by f and stride = s are applied and the result is $3 \times (m - f + 1) \times (n - f + 1)$ hidden feature neurons if a stride of 1 is used and the pooling layer result will be $3 \times (m - r + 1)/2 \times (n - r + 1)/2$ hidden features neurons when applied to 2×2 regions. The convolution operation generates the features map by multiplying the element of the input array by the element of the filter element wise and summing up the result to generate on pixel of the features map. Sliding the filter across the matrix and repeating the multiplication and summing up operations will generate the rest of features map pixels. The mathematical equation of this convolution operation is given as follows

$$O(i,j) = \sum_{k=1}^{f} \left(\sum_{l=1}^{f} input(i + k - 1, j + l - 1) kernel(k, l) \right) \tag{15}$$

where $i = 1, 2, \ldots m - f + 1$, $j = 1, 2, \ldots . . n - f + 1$.

1D-CNN is a simple CNN architecture that has only one convolutional layer. The simple design of this model leads to reduced number of parameters that can be adjusted during the training process therefore, it is highly needed in the genomic studies where it is difficult to collect large data to train a deep learning model that has very large number of parameters[53]. The one dimensional that we used in this study was constructed by Mostavi et al.[53] for predicting cancer tumor based on gene expression data. The architecture of the model when using LASSO as a feature selection technique is shown in Fig. 2.

Regularization with early stopping. We applied 1D-CNN with early stopping regularization to avoid over-fitting. The over-fitting is usually caused by training the model too much, making it pick up the noise as an essential part of the data instead of relying only on the training data. Such noise is normally unique to each training data. It can lead to high variance in the model estimates. On the other hand, too little training can result

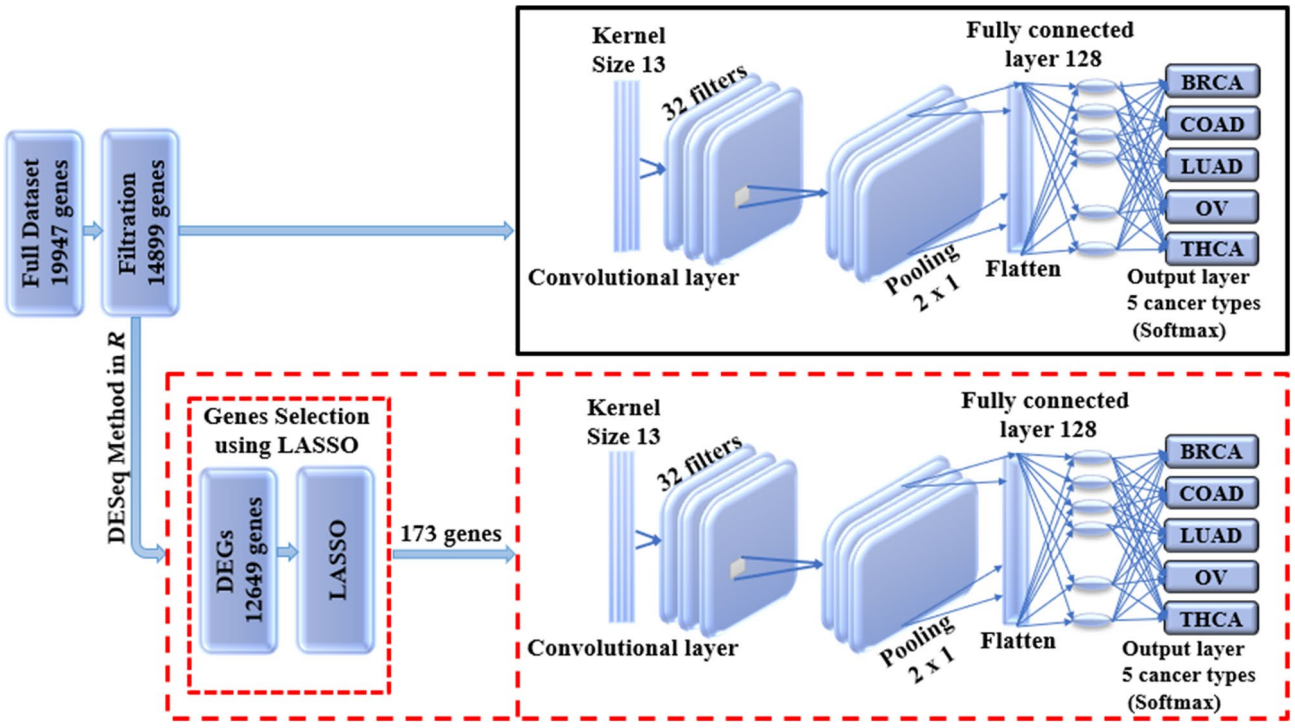


Figure 2. Illustrates the architecture of the 1D-CNN model. The upper panel presents the 1D-CNN without LASSO, while the lower panel shows the usage of LASSO as s feature selection technique for the 1D-CNN where it gives an input vector with 173 genes.

in under-fitting or high bias. Therefore, the variance and the bias have a negative relationship meaning that if the bias increases for fixed mean square error, then the variance will decrease and vice versa and that is known as the bias-variance tradeoff[54,55]. To avoid over-fitting, we can use a model with fewer parameters or obtaining more data. A model with fewer parameters can cause high bias. Since obtaining more data is not easy in the medical field, then a model with fewer parameters seems to be the alternative, but modern approaches in deep learning repeatedly show the benefits of using models with a large number of parameters[56,57]. Therefore, finding a way of adjusting the variance by minimizing noisy data can help solve the over-fitting problem. Since too much training can result in over-fitting, whereas too little training can result in under-fitting then the model can be regularized using the early stopping mechanism. We can implement the early stopping mechanism in the training procedure to make the architectures better fit the training data with each epoch and determining the number of epochs that can be run before the pre-trained model begin to overfit.

Stacking ensemble. Ensemble learning is the process of improving classifiers performance by combining the contribution of the trained sub-models to solve same classification problem[5]. Overall, each base learner votes and the final prediction is gained by the meta-learner, which is a model that learn to correct the prediction of the base-learners. Therefore, the ensemble approach results in prediction accuracy that is better than the single learners. Generalizability of an ensemble usually reduces the variance in the prediction, and thus ensure the most stable and best possible prediction is made. The meta model takes the output of the sub-models (base-learners) as input and then learns to merge the input prediction to make the final prediction which is better than each of the base-classifiers. Figure 3 shows our proposed stacking ensemble deep learning model.

Performance evaluation. We used different performance metrics to evaluate the performance of the classification methods. These metrics are namely accuracy, kappa, specificity, sensitivity, the area under the curve (AUC), precision, F-measure, and ROC curve. The accuracy measures the percentage of correctly classified cases but is not sufficient for measuring the performance of the classifier, especially if we have unbalanced data (which is the case with cancer data that we are dealing with). Sensitivity measures the percentage of the cases that are correctly classified as having cancer among those samples that are truly cancerous. Therefore, it measures the fraction of the correctly predicted cancer cases. Specificity measures the percentage of cases that do not have cancer, which are correctly identified to be so. In other words, it measures the true negative rate. Precision is the percentage of cases among those classified as positive that are truly positive, i.e., having cancer, and sometimes this measure is called the positive predicted value. F-measure is a measure that balances between precision and sensitivity.

We also compared the predictive performance of the methods using the receiver operating characteristic (ROC) curve plots. These figures were plotted using *MultiROC* package in *R*[58]. *MultiROC* calculates and visualizes ROC curve for multi-class using *micro-averaging* and *macro-averaging* approaches. *Micro-averaging* ROC-AUC converts the multi-class classification into binary classification by stacking all groups together. *Macro-averaging*

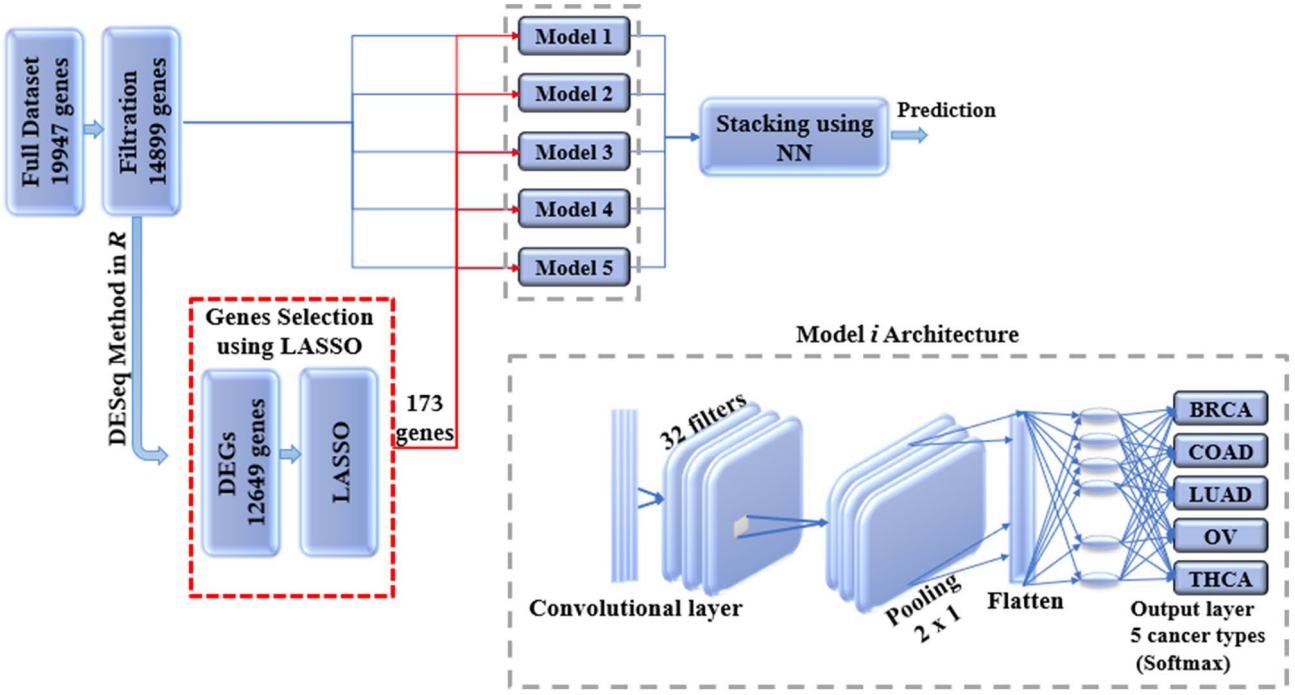


Figure 3. Stacking ensemble deep learning model architecture in which five 1D-CNN models are used as base models and the results of these models are combined using NN, which is used as a meta model. The NN has one hidden layer and an output layer that is activated using softmax function.

ROC-AUC uses one versus the rest approach by averaging all group's results and linear interpolation used between the points of the ROC. Confidence intervals for kappa statistics were computed using *vcd* package.

Methods to adjust for class imbalances. Imbalanced class sizes may lead to poor predictive performance particularly for the classes with small samples (Table 1). In order to handle the class imbalance and hence improve the models' performances we used the synthetic minority over-sampling technique (SMOTE) and under-sampling (DOWN) methods. SMOTE has been used widely in various fields such as bioinformatics for addressing the class imbalance in the outcome[59,60]. SMOTE is a data augmentation method that add new data to the minority class that are synthesized from the existing data instead of duplicating the data, because the duplication will not provide any new information to the model. SMOTE works by first selecting randomly a class instance *a* from the minority class then it chooses randomly one of the k nearest neighbors *b* to create the synthetic instances as a convex combination of *a* and *b* and finally, it forms a line segment in the feature space by connecting *a* and *b*.

We synthesized the minority class from existing samples by selecting randomly the closest *k* minority nearest neighbors to balance the class[61–63]. This statistical technique increases and generates the samples to reach the highest majority class and it makes the samples more general. SMOTE is implemented using *caret* package in *R* by adjusting the sampling method in the train control parameter to be 'SMOTE'.

Under-sampling technique (DOWN) tends to produce a new balanced subset of the original dataset by randomly removing instances usually from the majority class observations[64,65]. DOWN is implemented using *caret* package in *R* by adjusting the sampling method in the train control parameter to be 'DOWN'.

Statistical significance test. There are many different techniques that can used for comparing the accuracies of the machine learning models. In this work, we used the *resamples* method in *R* to analyze and visualize the estimated performance of the models. We used the *summary* function to compute summary statistics across each model/metric combination. *Diff* function in *R* is used to estimate the differences between the methods. The *diff* function performs a pairwise comparisons to compute the differences between pairs of consecutive elements using Bonferroni correction as an adjustment method. Bonferroni test is a type of multiple testing method used in statistical analysis to reduce the instance of a false positive and prevent the data from appearing incorrectly to be statistically significant[66,67].

Results

We found that the performance of the machine learning methods when LASSO as feature selection technique used is by far better than when it is not used. The performance of the methods in terms of overall statistics are summarized in Table 2 based on the under-sampling technique. Table 3 shows the results of methods in terms of per-class statistics for under-sampling technique. The receiver operating characteristic (ROC) curve plots

Methods	Performance measures					
	ACC (95% CI)	Kappa (95% CI)	F1-Score	Precision	Sensitivity	AUC
SVM-R	95.84 (94.00, 97.24)	93.81 (91.55, 96.07)	98.64	99.39	97.90	98.04
SVM-L	96.76 (95.10, 97.99)	95.14 (92.74, 97.18)	97.48	100.0	95.08	98.56
SVM-P	98.92 (97.79, 99.57)	98.40 (97.89, 99.74)	99.24	99.69	98.79	99.50
ANN	80.74 (77.49, 83.71)	72.15 (70.39, 79.59)	87.46	84.80	90.29	83.84
kNN	93.07 (90.83, 94.90)	89.97 (87.18, 92.75)	95.91	92.70	99.34	94.94
Bagging trees	99.20 (98.21, 99.75)	98.86 (97.86, 99.85)	99.54	99.69	99.39	99.54

Table 2. The overall predictive performance of the machine learning methods based on under-sampling. *SVM-R* support vector machine with radial-basis function (RBF) kernel, *SVM-L* support vector machine with linear kernel, *SVM-P* support vector machine with polynomial kernel, *ANN* Artificial Neural Networks, *kNN* K-nearest neighbors, Bagging trees; *ACC* accuracy, *CI* confidence interval, *Kappa* kappa statistics, *AUC* area under the curve.

Performance measures	Methods						
	Class	SVM-R	SVM-L	SVM-P	ANN	kNN	Bagging trees
Accuracy	BRCA	98.6	97.3	99.2	87.7	96.0	99.5
	COAD	95.8	98.6	98.6	90.2	94.7	98.5
	LUAD	97.7	99.6	98.0	82.8	90.6	98.7
	OV	90.7	88.5	98.9	93.4	98.5	100
	THCA	97.8	100	100	82.5	99.1	99.6
Sensitivity	BRCA	99.4	100	99.7	84.8	92.7	99.7
	COAD	91.7	97.2	97.2	86.1	94.4	97.2
	LUAD	98.8	100	96.5	68.6	81.4	97.7
	OV	81.6	77.0	97.7	92.0	98.9	100
	THCA	95.5	100	100	67.6	98.2	99.1
Specificity	BRCA	97.8	94.7	98.8	90.6	99.4	99.4
	COAD	100	100	100	94.3	94.9	99.8
	LUAD	96.6	99.3	99.5	97.0	99.8	99.6
	OV	99.8	100	100	94.8	98.0	100
	THCA	100	100	100	97.4	100	100
F1-score	BRCA	98.6	97.5	99.2	87.5	95.9	99.5
	COAD	95.7	98.6	98.6	60.8	67.3	97.2
	LUAD	89.5	97.7	96.5	72.8	89.2	97.7
	OV	89.3	87.0	98.8	81.6	93.5	100
	THCA	97.7	100	100	75.0	99.1	99.6
Precision	BRCA	97.9	95.1	98.8	90.3	99.4	99.4
	COAD	100	100	100	47.0	52.3	97.2
	LUAD	81.7	95.6	96.5	77.6	98.6	97.7
	OV	98.6	100	100	73.4	88.7	100
	THCA	100	100	100	84.3	100	100

Table 3. Predictive performance of the machine learning methods per-class statistics based on under-sampling. *SVM-R* support vector machine with radial-basis function (RBF) kernel, *SVM-L* support vector machine with linear kernel, *SVM-P* support vector machine with polynomial kernel, *ANN* Artificial Neural Networks, *kNN* K-nearest neighbors, Bagging trees, *ACC* Accuracy, *CI* confidence interval, *Kappa* kappa statistics *AUC* area under the curve.

comparing the machine learning classification methods in this study are shown in, Figs. 4, 5, 6, 7, 8 and 9 based on under-sampling method. The predictive performance of the under-sampling technique outperformed the over-sampling technique. Results for the over-sampling technique are available in the Supplementary File 2.

The overall predictive performance of the machine learning methods based on the under-sampling technique. The accuracy, precision, sensitivity, and F1-Score performance measures for the overall multi-class classification problem based on the under-sampling technique (DOWN) are presented in Table 2. These results show that bagging trees method achieved the best performance measure compared to the other

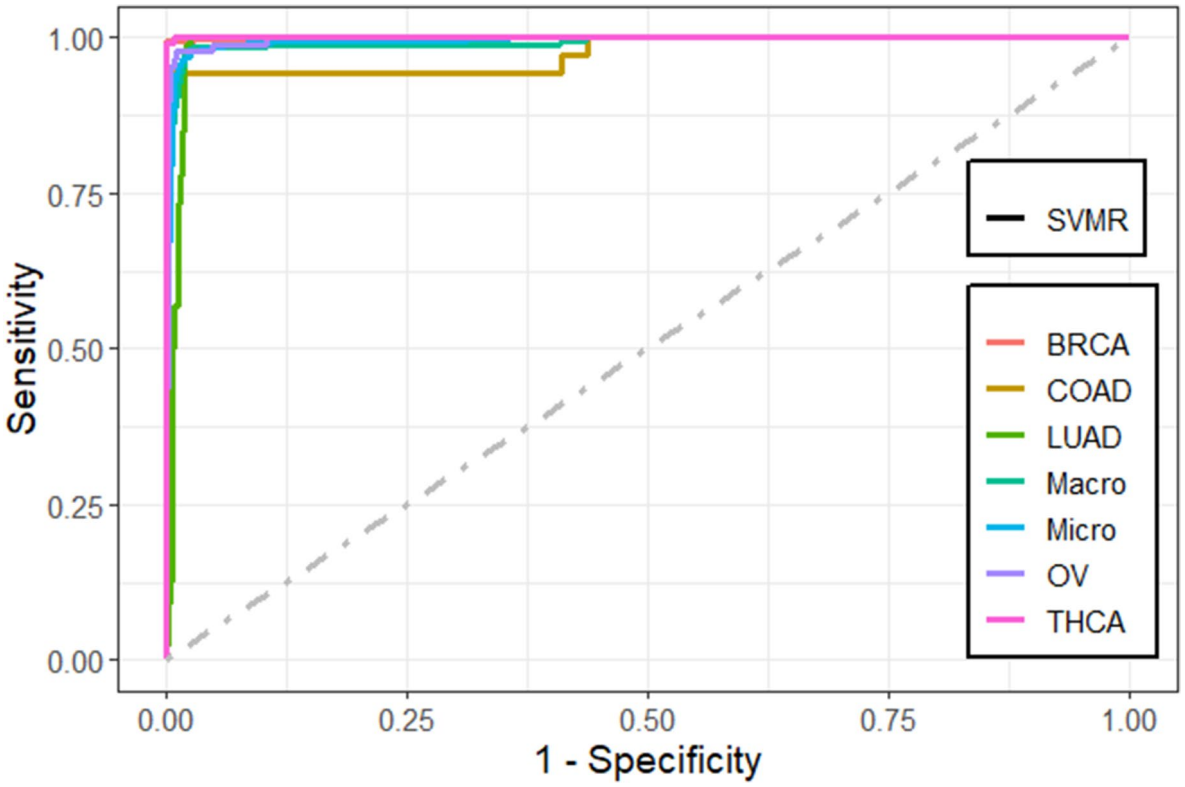


Figure 4. Multi-class ROC curves visualization for the SVMR model based on under-sampling technique.

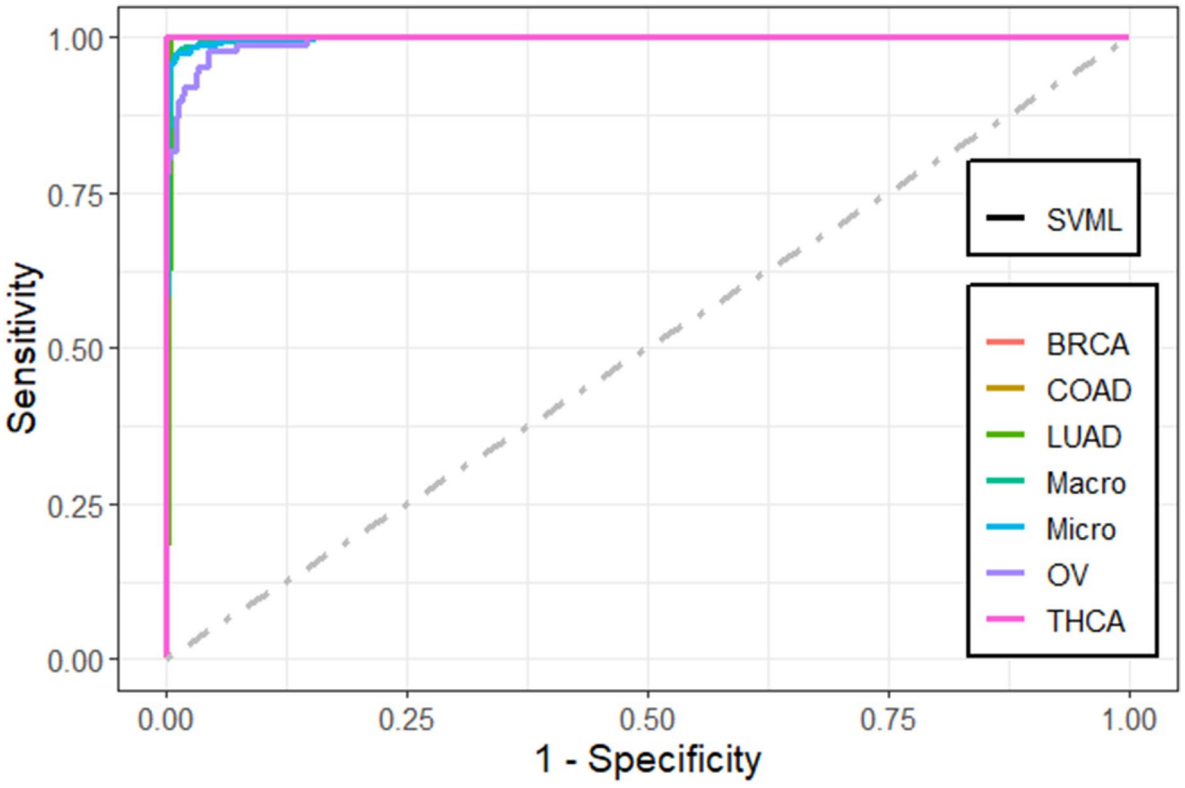


Figure 5. Multi-class ROC curves visualization for the SVML model based on under-sampling technique.

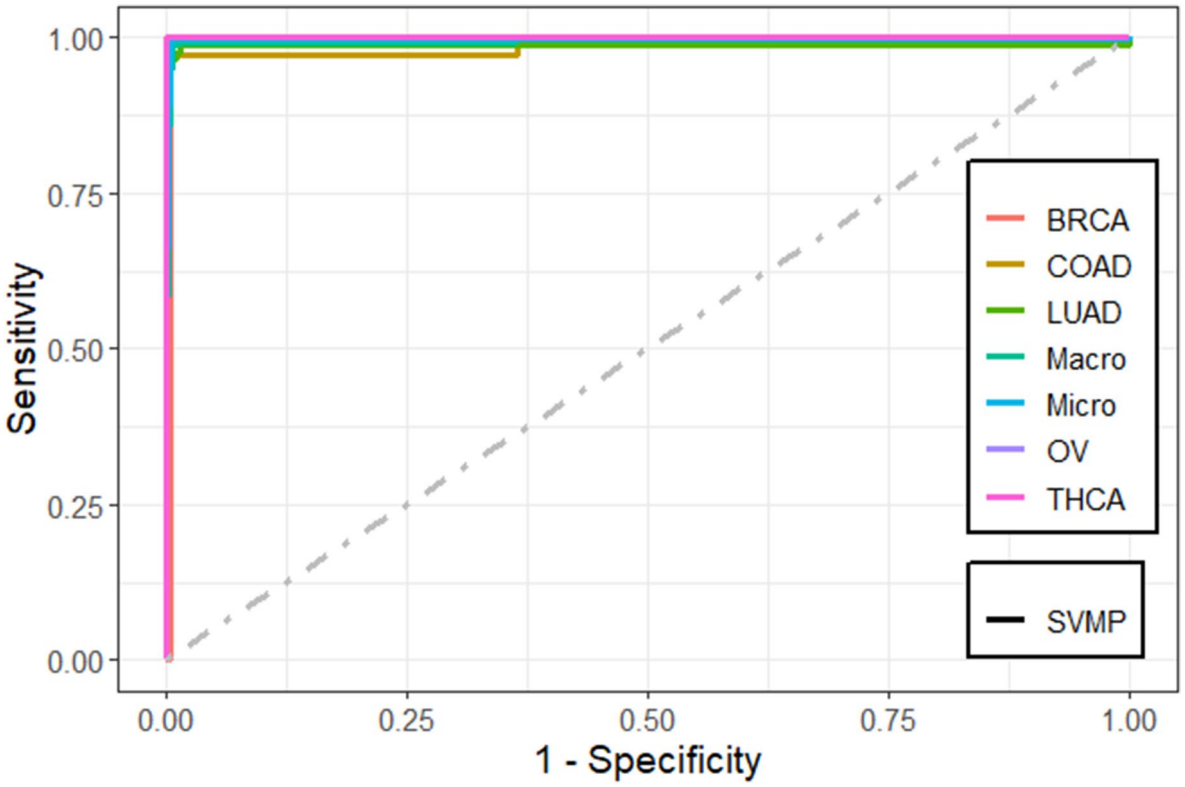


Figure 6. Multi-class ROC curves visualization for the SVMP model based on under-sampling technique.

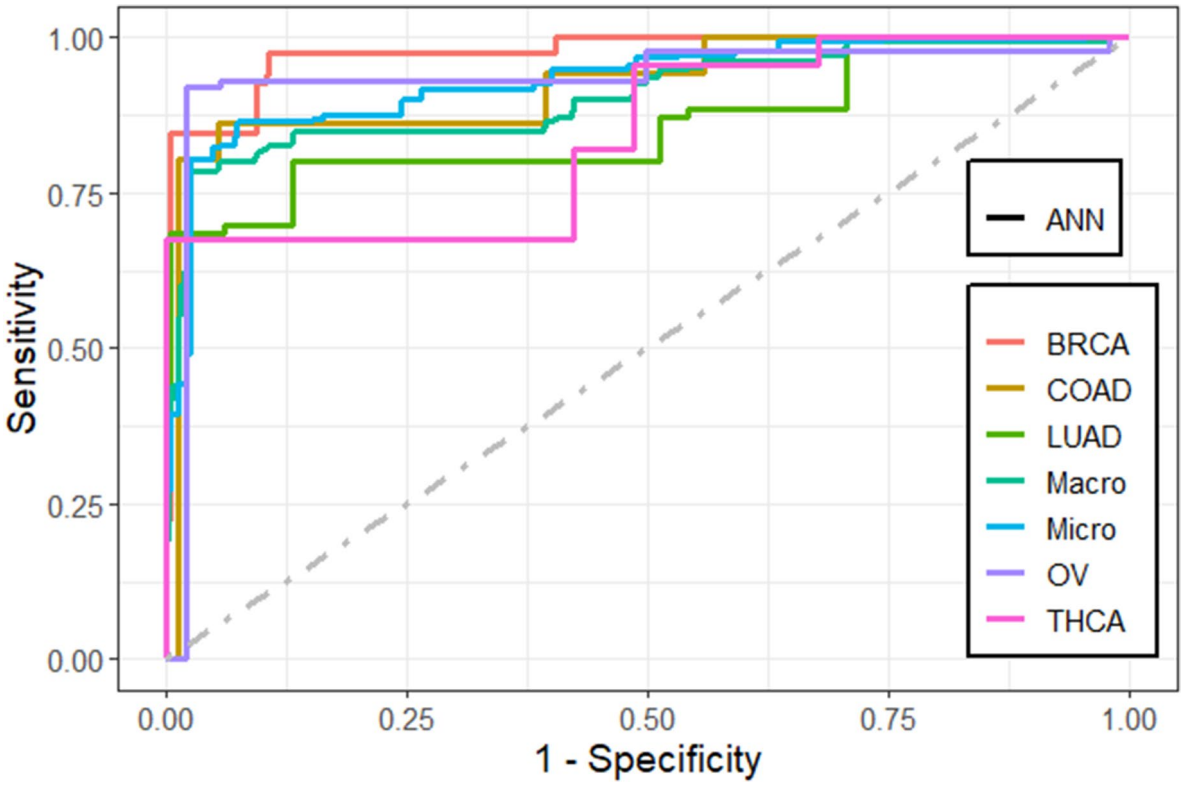


Figure 7. Multi-class ROC curves visualization for the ANN model based on under-sampling technique.

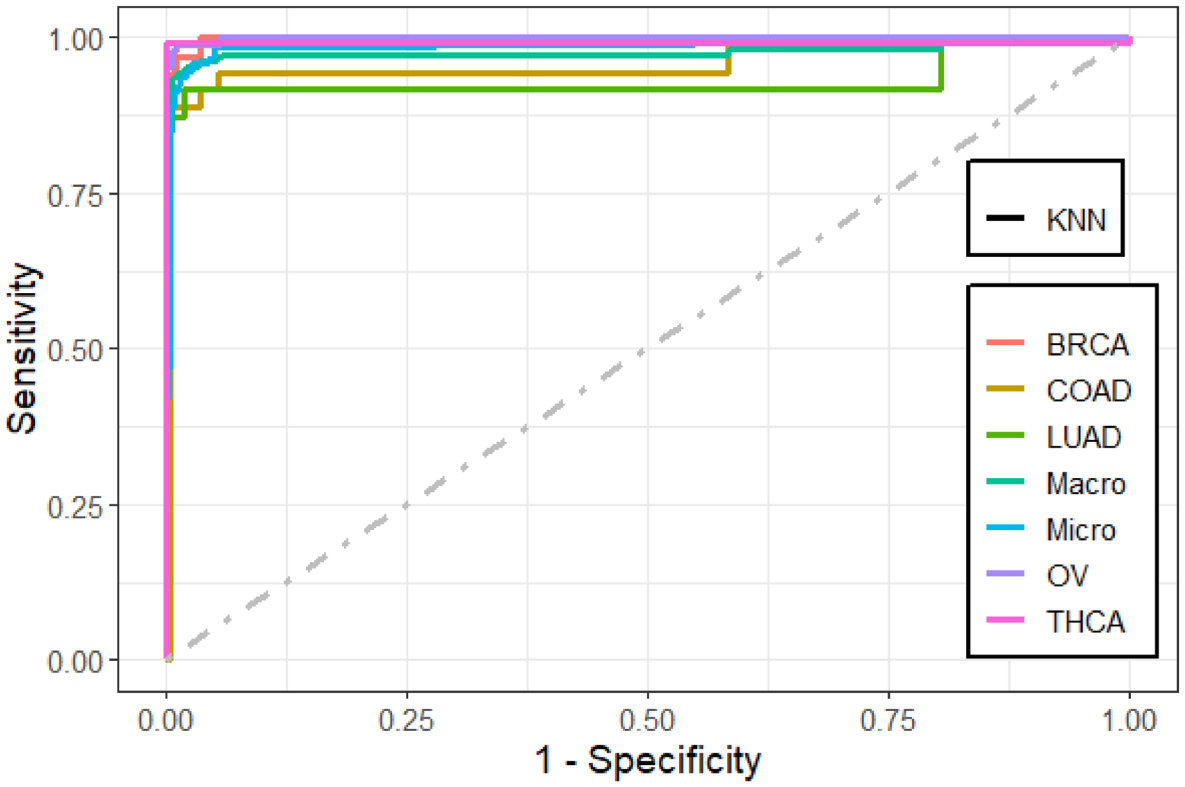


Figure 8. Multi-class ROC curves visualization for the KNN model based on under-sampling technique.

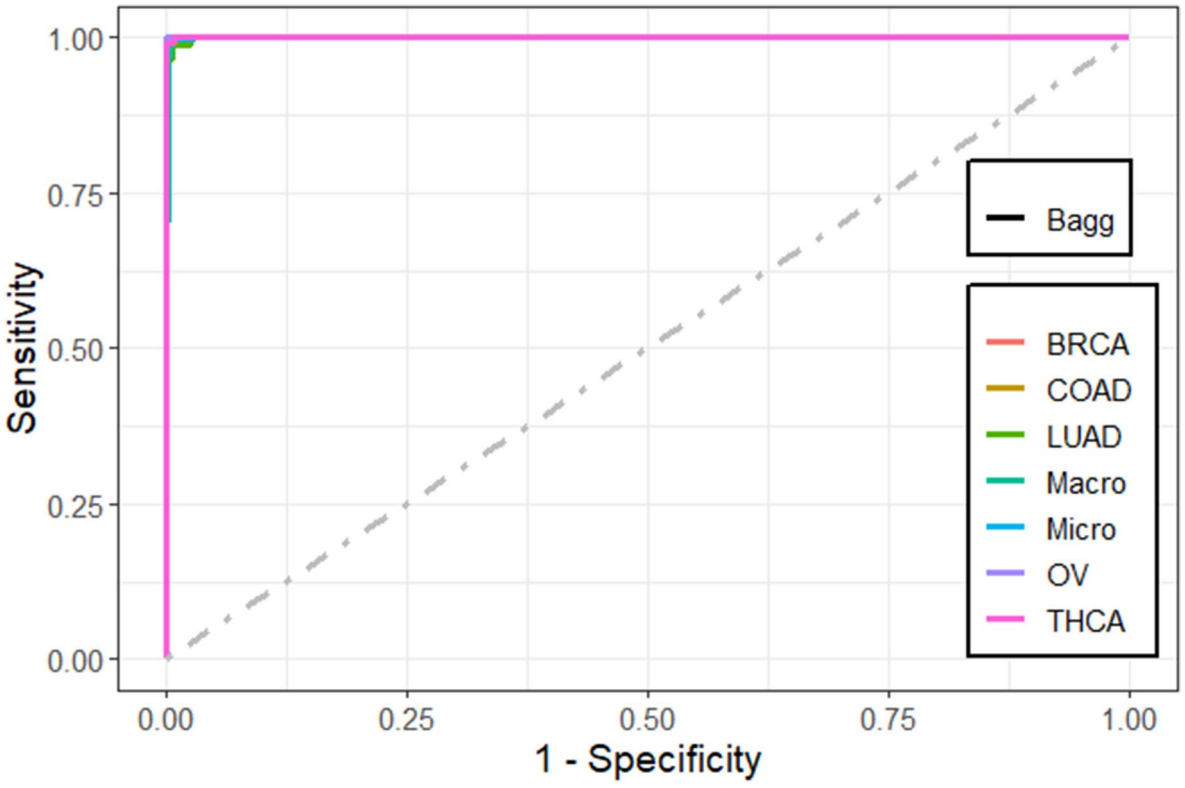


Figure 9. Multi-class ROC curves visualization for the bagging trees model based on under-sampling technique.

Performance measures	Folds										Overall
	1	2	3	4	5	6	7	8	9	10	
All (14,899 genes)											
Accuracy	99.54	98.16	95.85	97.24	97.24	97.24	99.54	96.30	99.54	100	98.06
Precision	99.47	96.07	93.50	96.72	96.92	95.11	99.82	94.16	99.38	100	97.12
Recall	99.26	98.20	96.56	95.22	96.82	96.06	99.26	94.94	99.81	100	97.61
F1-score	99.36	97.03	94.87	95.94	96.78	95.48	99.53	94.54	99.59	100	97.31
Reduced (173 genes)											
Accuracy	98.62	99.54	99.08	98.62	99.54	100	99.07	99.54	98.61	99.54	99.22
Precision	99.46	99.31	99.10	98.99	99.82	100	98.48	99.29	98.92	99.82	99.32
Recall	97.97	99.82	99.10	98.39	99.29	100	98.72	99.81	98.52	99.26	99.09
F1-score	98.68	99.56	99.10	98.65	99.54	100	98.57	99.54	98.69	99.53	99.19

Table 4. The performance of the 1D-CNN model using early stopping regularization.

methods where it yields an accuracy, sensitivity, AUC, and F1-score of 99.2%, 99.4%, 99.54, and 99.5%, respectively. However, SVM-P and bagging trees have the same precision, and they have a close results in the other performance measures. Consequently, ANN method obtained the worst performance with an accuracy of 80.7%.

Predictive performance of the machine learning methods per cancer tumor based on the under-sampling. The accuracy, precision, sensitivity, and F1-Score performance measures based on per-class statistics using the under-sampling technique method (DOWN) are presented in Table 3. Bagging trees outperforms the other methods in classifying most of the five cancer tumors in most of the performance measures, followed by SVM-P method. While the ANN shows the lowest performance measures. These results were confirmed using the ROC curves which are depicted in Figs. 4, 5, 6, 7, 8, and 9. Bagging trees was able to highly correctly classify the ovarian cancer with 100% in terms of accuracy, sensitivity, specificity, F1-Score, and precision. While SVM-L and SVM-P can sensitively classify the thyroid cancer with a 100% of accuracy, sensitivity, specificity, F1-Score, and precision. Also, SVM-R shows performance that is close to SVM-L and SVM-P to classify the thyroid cancer.

Predictive performance of the one-dimensional convolutional neural network model. The results that are presented in Table 4 show that the 1D-CNN model has a high performance when applied on the genes that are selected using LASSO (173 genes) where it achieved an average classification accuracy of 99.22%. These results also showed that the 1D-CNN outperformed the results of the machine learning methods that are presented in Table 2. It can be noted from the overlapped confusion matrix of the multiclass classification that the deep learning model classified the five categories of the cancers types using the 173 genes better than classifying these categories using the full list of genes (14,899). The resulting precision, recall, and F1-score values are 99.32%, 99.09%, and 99.19%, respectively.

Figures 10, 11, 12 and 13 show F1-measure and accuracy for training and validation when training our model using the full list of genes and the reduced genes with the early stopping approach. These figures indicate that the model can generalize very well since they become stable when the F1-measure and the accuracy are more than 99%. Figures 14 and 15 show the losses when using the full list of genes and the LASSO selected genes, respectively.

The multi-class classification performance of the 1D-CNN model has been evaluated for each fold, and the average classification performance of the model is calculated. The overlapped confusion matrix (CM) is shown in Figs. 16 and 17 for all and reduced lists of genes, respectively. The overlapped CM is created using the sum of the ten separated confusion matrices. Thus, it is aimed to obtain an idea about the general perforations of the model.

Although we are using RNAseq data with a high number of genes, deep learning method outperformed the machine learning methods noting that a rigorous preprocessing step including a model-based approach using LASSO regression was applied to reduce the number of genes to be less than the number of observations.

The results that are presented in Table 5 below show that our proposed model has a high performance when applied on the genes that are selected using LASSO (173 genes) where it achieved an average precision, recall, and F1-Score of 99.55, 99.29, and 99.42 respectively. While the classification accuracy is 99.45% which is lower compared to accuracy of the full genes. These results also showed that our proposed model outperformed the results of the single 1D-CNN model and machine learning that are presented in Tables 2 and 4. In addition, Figs. 18 and 19 which is the overlapped confusion show that our proposed model has a better classification performance compared compared to the single 1D-CNN. Overall, our proposed model performance without using LASSO as a feature selection method is comparable to the performance with LASSO.

A comparison of the methods was statistically conducted using the pairwise analysis test which produced pairwise statistical significance table of scores where the lower diagonal of the table shows p-values for the null hypothesis (distributions are the same), smaller p-value is indicative of a better model. The upper diagonal of the table presents the estimated differences in mean accuracy and kappa coefficient between the distributions. From Table 6 (under-sampling technique) we can see clearly of the fifteen pairwise comparisons of the six

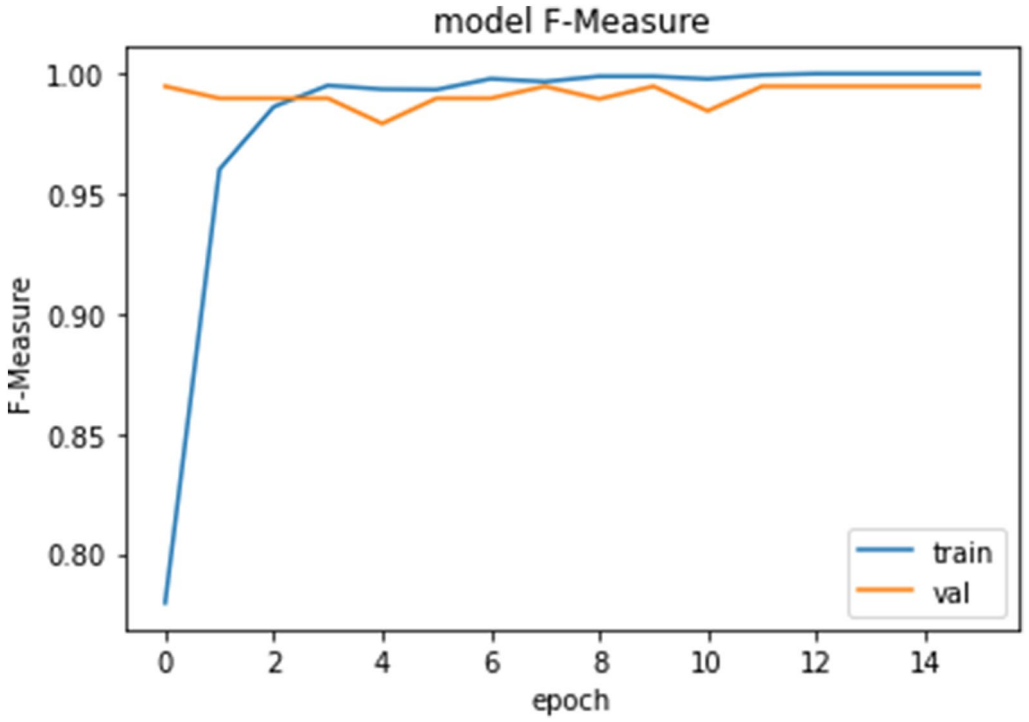


Figure 10. Training and validation F1 measure for the full list of genes with early stopping.

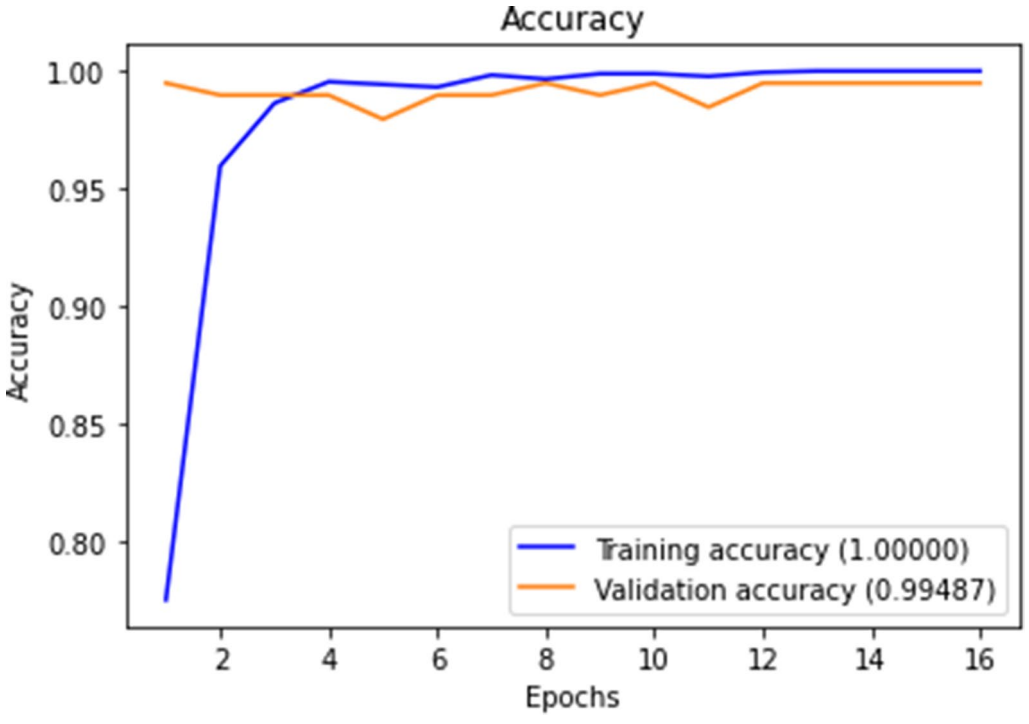


Figure 11. Training and validation accuracy for the full list of genes with early stopping.

machine learning methods, there are nine comparisons showing statistically significant differences in terms of accuracy at the 0.05 level of significance. These differences are SVMR differed statistically to SVMP $p = 0.003$, ANN $p = < 0.001$, and KNN $p = < 0.001$. While SVML differed statistically to ANN $p = 0.009$, and SVMP differed statistically to ANN $p = < 0.001$ and KNN $p = < 0.001$. Moreover, ANN differed statistically to bagging trees $p = < 0.001$, as well as KNN differed statistically to bagging trees $p = 0.004$.

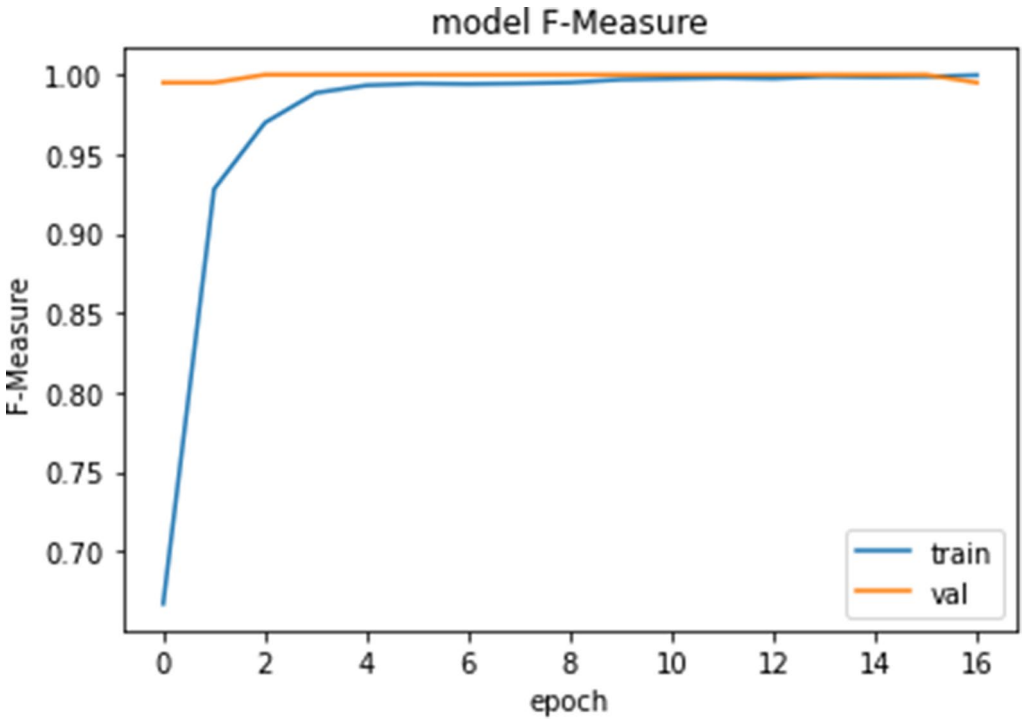


Figure 12. Training and validation F1 measure for reduced genes with early stopping.

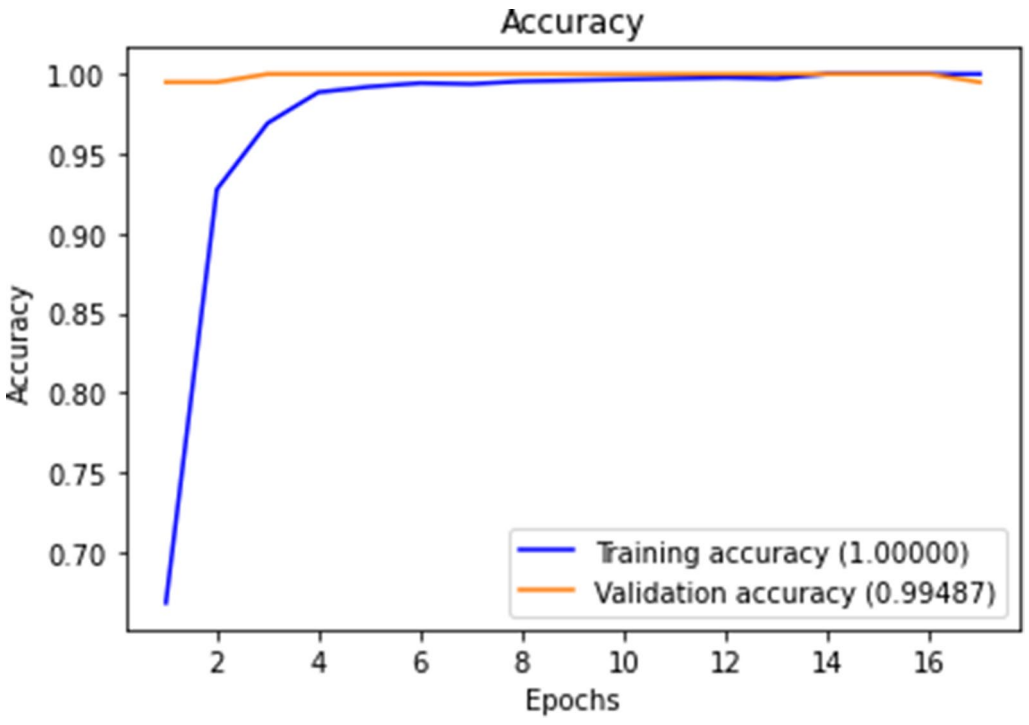


Figure 13. Training and validation accuracy for reduced genes with early stopping.

Discussion

We applied a novel stacking ensemble deep learning model to classify five common cancers among women: breast, colon adenocarcinoma, lung adenocarcinoma, ovarian, and thyroid cancers. The performance of the current proposed model is compared with the single 1D-CNN and machine learning methods that are mostly used in cancer types classification. We showed that the best machine learning average results were obtained using 173 genes based on the under-sampling technique, while our proposed model has the highest performance

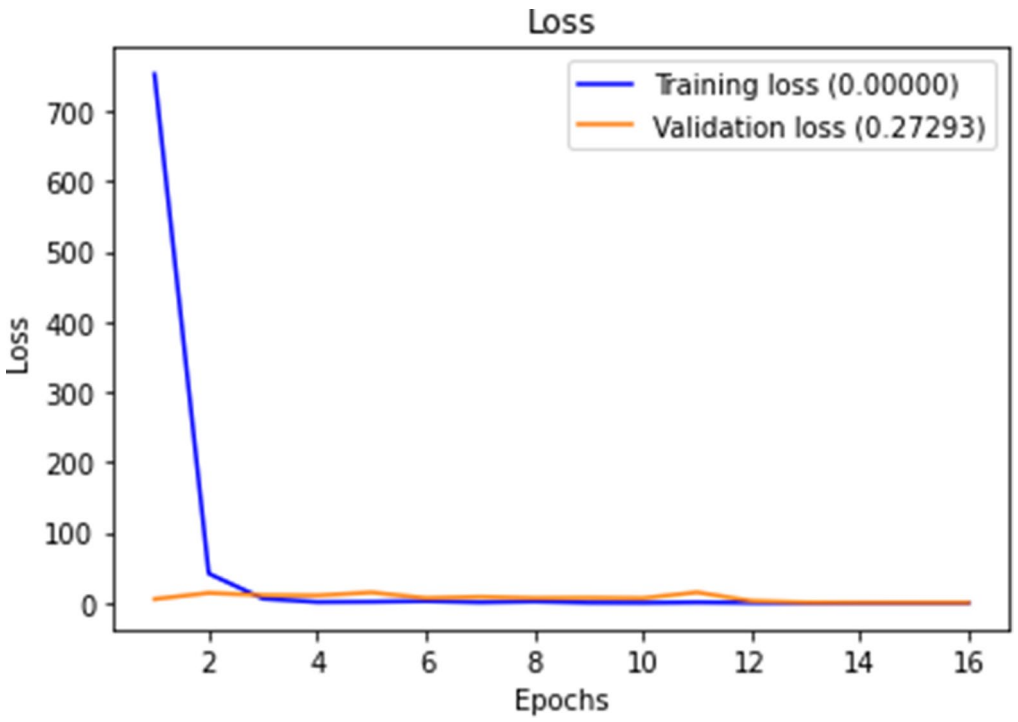


Figure 14. Training and validation loss for the full list of genes with early stopping.

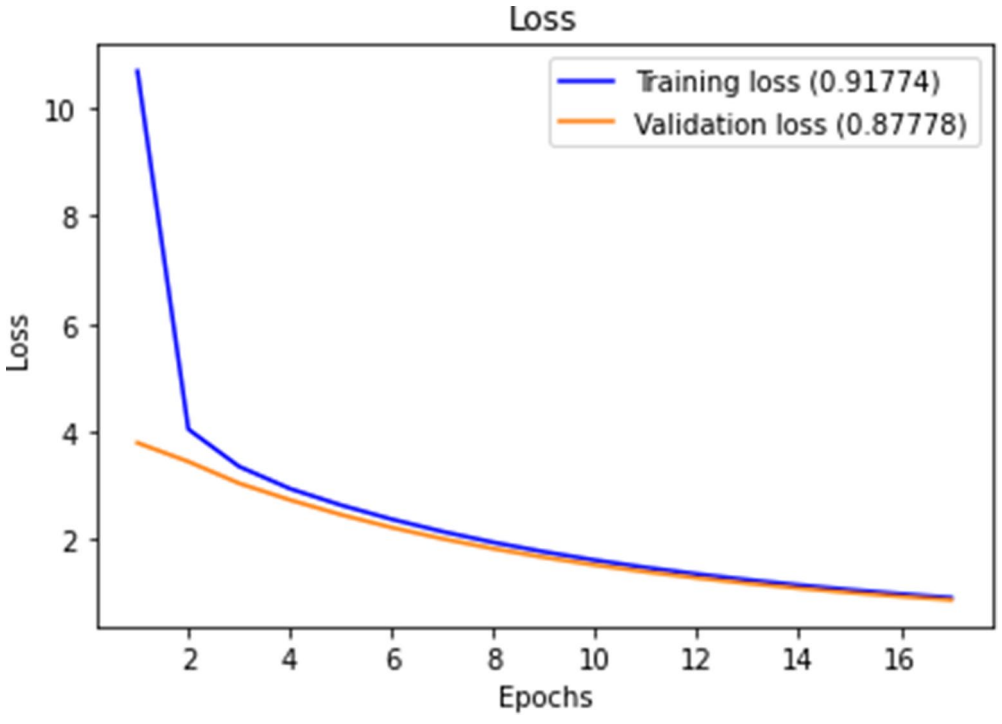


Figure 15. Training and validation loss for reduced genes with early stopping.

based on the early stopping regularization. The improvement in accuracy was achieved by optimizing several parameters. We used LASSO as a feature selection technique with our proposed model to explore the integration of features selection method with a deep learning approach because features selection in deep learning is still unexplored area due to the black box nature of the deep learning methods. The results of the proposed model without using LASSO as a feature selection technique is comparable to the results with LASSO. This indicates that the 1D-CNN performs features selection through its layers. Bagging trees obtained excellent results, with a

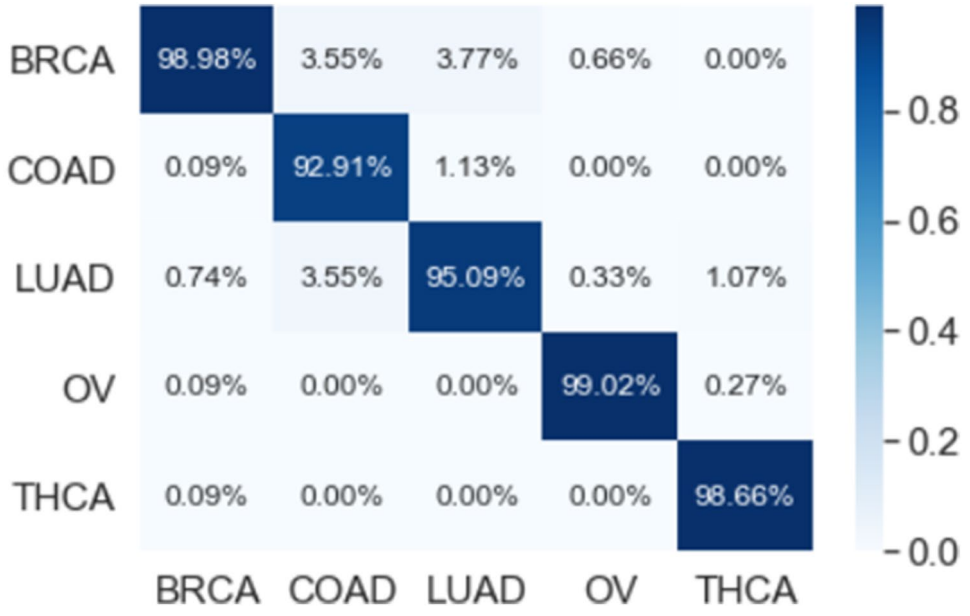


Figure 16. 10-folds overlapped confusion matrix (CM) for all 14,899 genes.

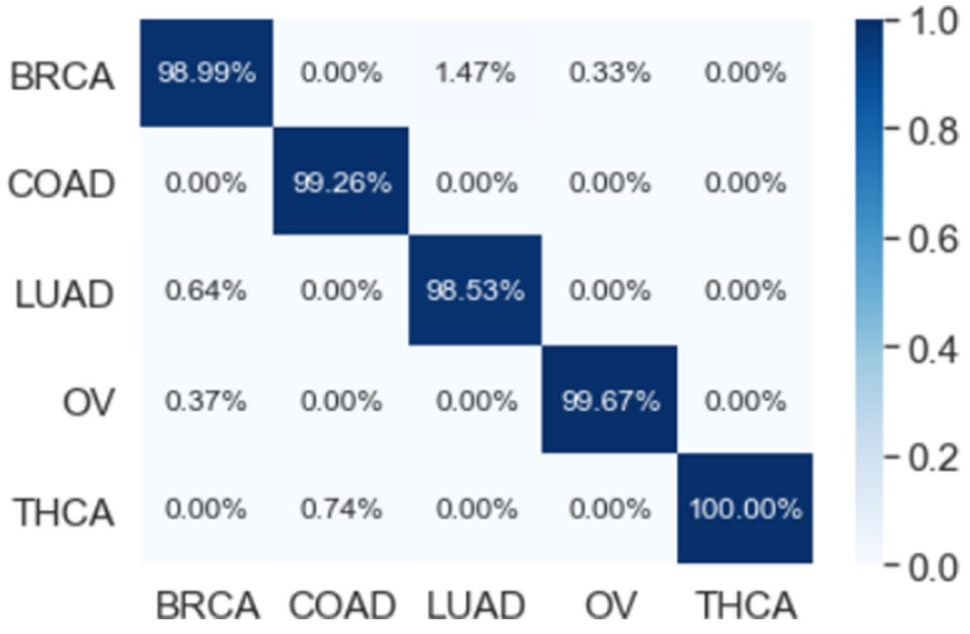


Figure 17. 10-folds overlapped confusion matrix (CM) for the reduced 173 genes.

maximum accuracy of 99.2% among the machine learning models based on the under-sampling technique. In contrast, ANN showed the least accuracy of 80.7% for classifying the most common cancers among females. The SVM-P method showed performances that was close to the bagging trees method with an accuracy of 98.9% when we used the under-sampling technique. Overall, our results showed that SVM-R, SVM-L, SVM-P, ANN, KNN, and bagging trees were improved in performance if under-sampling is applied compared to over-sampling. We conclude that our proposed model is the best methods for the test dataset in this study. However, bagging trees is the best model among the machine learning models.

Overall, our proposed model surpassed the single 1D-CNN and the machine learning methods in the classification of common cancers among women. These findings are different from those reported in other studies[11,18,19]. These differences can be explained by variations in the type of cancers studied and the methods used for feature/gene selection. A study by Yang and Naiman[14] introduced and validated a gene selection approach using machine learning methods but did not assess the performance of the machines. Our findings demonstrated that, our

Performance measures	Folds										Overall
	1	2	3	4	5	6	7	8	9	10	
All (14,899 genes)											
Accuracy	99.45	99.26	99.63	99.08	99.63	99.45	99.63	99.45	99.63	99.63	99.48
Precision	99.23	99.15	99.57	98.57	99.57	99.23	99.57	99.23	99.57	99.57	99.33
Recall	98.88	98.53	99.57	98.12	99.57	99.50	99.57	98.88	99.57	99.57	99.18
F1-score	99.05	98.83	99.57	98.31	99.57	99.36	99.57	99.05	99.57	99.57	99.25
Reduced (173 genes)											
Accuracy	99.45	99.26	99.26	99.26	99.45	99.45	99.45	99.63	99.82	99.45	99.45
Precision	99.58	99.31	99.13	99.31	99.58	99.60	99.58	99.65	99.93	99.79	99.55
Recall	99.19	99.12	99.31	99.12	99.19	99.38	99.19	99.47	99.72	99.19	99.29
F1-score	99.38	99.22	99.22	99.22	99.38	99.49	99.38	99.56	99.82	99.49	99.42

Table 5. The performance of the new proposed model using early stopping regularization.

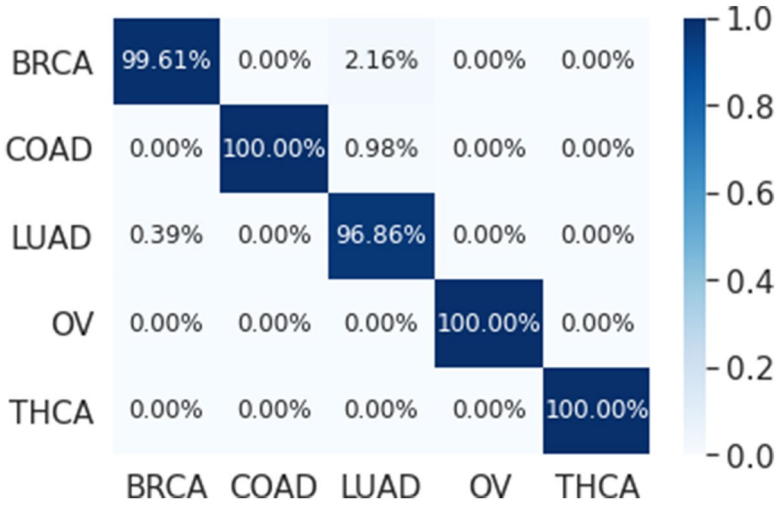


Figure 18. 10-folds stacking ensemble deep learning model overlapped confusion matrix (CM) for all 14,899 genes.

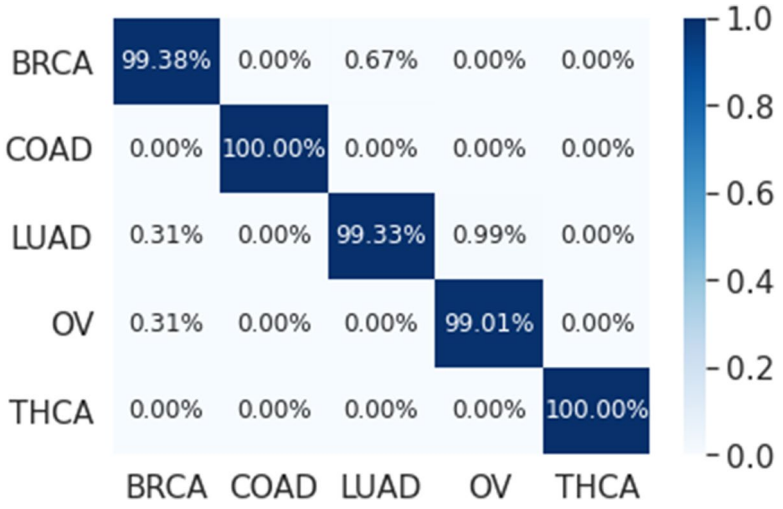


Figure 19. 10-folds stacking ensemble deep learning model overlapped confusion matrix (CM) for the reduced 173 genes.

	SVMR	SVML	SVMP	ANN	KNN	Bagging trees
Accuracy						
SVMR		0.015	−0.015	0.138	0.038	−0.003
SVML	1.00		−0.030	0.123	0.022	−0.019
SVMP	0.003	0.347		0.153	0.052	0.011
ANN	< 0.001	0.009	< 0.001		−0.101	−0.142
KNN	< 0.001	1.00	< 0.001	0.008		−0.041
Bagging trees	1.00	1.00	0.250	< 0.001	0.004	
Kappa						
SVMR		0.024	−0.021	0.194	0.054	−0.005
SVML	1.00		−0.045	0.170	0.030	−0.029
SVMP	0.003	0.386		0.215	0.075	0.016
ANN	< 0.001	0.010	< 0.001		−0.140	−0.199
KNN	< 0.001	1.00	< 0.001	0.006		−0.059
Bagging trees	1.00	1.00	0.250	< 0.001	0.004	

Table 6. Pairwise statistical analysis test p-values and the estimated differences for the machine learning models (under-sampling technique).

proposed model can achieve a higher performance on cancer tumor classification using gene expression data. Both deep and machine learning methods and a combination of both can assist in predicting or detecting cancer susceptibility in the early stages and therefore, aid in designing early treatment strategies, and in turn increase survival of the high-risk women.

Because of the large number of genes in the gene expression data, we used LASSO regression as a rigorous feature selection method that reduced the dimensionality of the data sets[24,68]. This process enabled us to retain the most important features (genes) for classification and prediction. In order to avoid over-fitting and the bias in the skewed class distribution we used over and under-sampling imbalance handling techniques, which improve the machine learning performance. In general, our results show that under-sampling technique improved the methods performance, and this is confirmed in previous studies[64,65,69].

There were statistically significant differences ($p < 0.05$) between the machine learning methods, which demonstrates that the performance of the machines on cancer classification is not the same. However, deep learning methods outperformed the machine learning methods in cancer classification, which is similar to a previous study[23]. Overall, the accuracy of our proposed model on the full features and on the features that are selected using LASSO are 99.48% and 99.45, respectively, which are 5.05% and 5.02% higher than accuracy obtained by[23] which is 94.43%. We note that Tabares-Soto et al.[24] used microarray gene expression data, focusing on 11 type of cancers for both males and females, compared to RNASeq data used in this study to classify five common cancers among females. This study also did not consider class imbalance handling methods as applied in the current study and had 12-times lower sample size (n = 174) than in our study (n = 2166). With larger sample size, more samples are available to train the models. These issues were, therefore, likely to affect the reliability of findings and potentially affecting the performance of the methods. Our study was limited to the gene expression profiles from RNASeq data. However, Lee and co-workers[22] used several features such as mutation profiles and mutations rates. They evaluated different machine learning and feature selection methods using RNASeq data from 31 cancer types. The highest accuracy they obtained was 84%. Thereafter, they reduced the number of cancers to the six most common types and obtained an accuracy of 94%, which is low compared to our proposed deep learning model.

Our proposed model has a very high achievement in classifying the five common cancers among women and that may potentially improve the multi-class identification[19]. In addition, this study is first of its kind to classify cancer tumors using RNAseq data. However, multi-class cancer classification using gene expression is not a substitute to the traditional diagnosis[19], but advances in classification algorithms or methods may provide a more accurate and biologically meaningful classifications and inform future studies. Moreover, a more pressing classification problem may be that of discriminating between cancer sub-types within the same type than between cancer types. However, we postulate that the methods covered in this paper are directly applicable to this problem.

Conclusion

In this work, we proposed a stacking ensemble deep learning model as a multi-class classifier to classify five most common cancers among women, that is, breast, colon adenocarcinoma, lung adenocarcinoma, ovarian, and thyroid cancer, using RNASeq gene expression datasets for each cancer tumor. Tumor classification using RNASeq data is more accurate and available compared to microarray data. We used LASSO as a feature selection method and compared the performance of our proposed method with a stand alone deep learning and machine learning methods. We conclude that our proposed model achieved the highest performance compared to the single 1D-CNN and the machine learning methods. Our proposed model is, therefore, capable of correctly classifying all the observed positive cancer cases. The proposed model can help improve the detection and diagnosis of cancer susceptibility among women in the early stages, inform decision on early intervention, and hence improve survival. Future research should consider the potential effects of using many feature types such as

methylations and mutations, to be integrated with RNASeq data. Future work will also consider improvements on the stacking ensemble problem including statistical properties to improve inference.

Data availability

The datasets are publicly available on The Cancer Genome Atlas (TCGA) repository.

Received: 25 November 2020; Accepted: 19 July 2021
Published online: 02 August 2021

References

1. Olsen M. Cancer in Sub-Saharan Africa: The need for new paradigms in global health: Boston University Frederick S. Pardee Center for the Study of the Longer, (2015).
2. Morhason-Bello, I. O. *et al.* Challenges and opportunities in cancer control in Africa: A perspective from the African Organisation for Research and Training in Cancer. *Lancet Oncol.* **14**(4), e142–e151 (2013).
3. Sung, H., Ferlay, J., Siegel, R. L., Laversanne M., Soerjomataram, I., Jemal, A., *et al.* Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA Cancer J. Clin. Epub 2021/02/05. https://doi.org/10.3322/caac.21660. PubMed PMID: 33538338, (2021).
4. Golub, T. R. *et al.* Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* **286**(5439), 531–537 (1999).
5. Mohammed, M., Mwambi, H., Omolo, B., & Elbashir, M. K. (eds.) Using stacking ensemble for microarray-based cancer classification. In *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE),* IEEE, (2018).
6. Tan, A. C. & Gilbert, D. Ensemble machine learning on gene expression data for cancer classification. *Appl. Bioinform.* **2**(3 Suppl), S75-83 (2003) (**Epub 2004/05/08 PubMed PMID: 15130820**).
7. Datta, S. & Nettleton, D. *Statistical Analysis of Next Generation Sequencing Data* (Springer, 2014).
8. Rai, M. F., Tycksen, E. D., Sandell, L. J. & Brophy, R. H. Advantages of RNA-seq compared to RNA microarrays for transcriptome profiling of anterior cruciate ligament tears. *J. Orthopaedic Res.* **36**(1), 484–497 (2018).
9. Koch, C. M. *et al.* A beginner's guide to analysis of RNA sequencing data. *Am. J. Respir. Cell Mol. Biol.* **59**(2), 145–157 (2018).
10. Zhao, S., Zhang, B., Zhang, Y., Gordon, W., Du, S., Paradis, T. *et al.* Bioinformatics for RNA-Seq Data Analysis. Bioinformatics—Updated Features and Applications. InTech, 125–149, (2016).
11. García-Díaz, P., Sánchez-Berriel, I., Martínez-Rojas, J. A. & Diez-Pascual, A. M. Unsupervised feature selection algorithm for multiclass cancer classification of gene expression RNA-Seq data. *Genomics* **112**(2), 1916–1925 (2020).
12. Abusamra, H. A comparative study of feature selection and classification methods for gene expression data of glioma. *Proc. Comput. Sci.* **23**, 5–14 (2013).
13. Torre, L. A., Islami, F., Siegel, R. L., Ward, E. M. & Jemal, A. Global cancer in women: burden and trends. *Cancer Epidemiol. Prevent. Biomark.* **26**(4), 444–457 (2017).
14. Yang, S. & Naiman, D. Q. Multiclass cancer classification based on gene expression comparison. *Stat. Appl. Genet. Mol. Biol.* **13**(4), 477–496 (2014).
15. Lusa, L. Class prediction for high-dimensional class-imbalanced data. *BMC Bioinform.* **11**(1), 523 (2010).
16. Ca, D. A. V. & Mc, V. Gene expression data classification using support vector machine and mutual information-based gene selection. *Proc. Comput. Sci.* **47**, 13–21 (2015).
17. Haury, A.-C., Gestraud, P. & Vert, J.-P. The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. *PLoS ONE* **6**(12), e28210 (2011).
18. Castillo, D. *et al.* Leukemia multiclass assessment and classification from Microarray and RNA-seq technologies integration at gene expression level. *PLoS ONE* **14**(2), e012127 (2019).
19. Ramaswamy, S. *et al.* Multiclass cancer diagnosis using tumor gene expression signatures. *Proc. Natl. Acad. Sci.* **98**(26), 15149–15154 (2001).
20. Nawaz, M., Sewissy, A. A. & Soliman, T. H. A. Multi-class breast cancer classification using deep learning convolutional neural network. *Int. J. Adv. Comput. Sci. Appl.* **9**(6), 316–332 (2018).
21. Piao, Y., Piao, M. & Ryu, K. H. Multiclass cancer classification using a feature subset-based ensemble from microRNA expression profiles. *Comput. Biol. Med.* **80**, 39–44 (2017).
22. Lee, K., Jeong, H.-O., Lee, S. & Jeong, W.-K. CPEM: Accurate cancer type classification based on somatic alterations using an ensemble of a random forest and a deep neural network. *Sci. Rep.* **9**(1), 1–9 (2019).
23. Tabares-Soto, R. *et al.* A comparative study of machine learning and deep learning algorithms to classify cancer types based on microarray gene expression data. *PeerJ Comput. Sci.* **6**, e270 (2020).
24. Fonti, V. & Belitser, E. Feature selection using lasso. *VU Amsterdam Res. Paper Business Anal.* **30**, 1–25 (2017).
25. Weinstein, J. N. *et al.* The cancer genome atlas pan-cancer analysis project. *Nat. Genet.* **45**(10), 1113 (2013).
26. Colaprico, A. *et al.* TCGAbiolinks: An R/Bioconductor package for integrative analysis of TCGA data. *Nucl. Acids Res.* **44**(8), e71 (2016).
27. Team RC. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing, (2020).
28. Bullard, J. H., Purdom, E., Hansen, K. D. & Dudoit, S. Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinform.* **11**(1), 94 (2010).
29. Anders, S., & Huber, W. Differential expression analysis for sequence count data. Genome. Biol. **11** (10) R106. Epub 2010/10/29. https://doi.org/10.1186/gb-2010-11-10-r106. PubMed PMID: 20979621; PubMed Central PMCID: PMCPMC3218662, (2010).
30. Michael, I., Love, S. A., Vladislav K., & Wolfgang H. RNA-seq workflow: gene-level exploratory analysis and differential expression: Bioconductor; 16 October, 2019 [cited 2020 May 1, 2020]. Available from: https://bioconductor.org/packages/release/workflows/vignettes/rnaseqGene/inst/doc/rnaseqGene.html#differential-expression-analysis.
31. Dündar, F., Skrabanek, L., & Zumbo, P. Introduction to differential gene expression analysis using RNA-seq. Appl Bioinform, 1–67 (2015).
32. Friedman, J., Hastie, T. & Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.* **33**(1), 1 (2010).
33. Pereira, J. M., Basto, M. & da Silva, A. F. The logistic lasso and ridge regression in predicting corporate failure. *Proc. Econ. Finance* **39**, 634–641 (2016).
34. Hastie, T., & Qian, J. An Introduction to glmnet. (2016).
35. Hu, H., Li, J., Plank, A., Wang, H., & Daggard, G. (eds.) A comparative study of classification methods for microarray data analysis. In *Proceedings of the 5th Australasian Data Mining Conference (AusDM 2006): Data Mining and Analytics*, (ACS Press, 2006).
36. Boser, B. E., Guyon, I. M., & Vapnik, V. N. (eds.) A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory,* (1992).

37. Brown, M. P., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C., Ares, M., *et al*. Support vector machine classification of microarray gene expression data. University of California, Santa Cruz, Technical Report UCSC-CRL-99-09. (1999).
38. Chu, F., & Wang, L. (eds.) Gene expression data analysis using support vector machines. In *Proceedings of the International Joint Conference on Neural Networks*, IEEE, (2003).
39. Muñoz, A., de Diego, I. M., & Moguerza, J. M. Support vector machine classifiers for asymmetric proximities. In *Artificial Neural Networks and Neural Information Processing—ICANN/ICONIP*, Springer 217–224 (2003).
40. Stephens, D. & Diesing, M. A comparison of supervised classification methods for the prediction of substrate type using multibeam acoustic and legacy grain-size data. *PLoS ONE* **9**(4), e93950 (2014).
41. Karatzoglou, A., Smola, A., Hornik, K., & Karatzoglou, M. A. Package 'kernlab'. Technical report, CRAN, 03 2016, (2019).
42. Dwivedi, A. K. Artificial neural network model for effective cancer classification using microarray gene expression data. *Neural Comput. Appl.* **29**(12), 1545–1554 (2018).
43. Lek, S., & Park Y. Artificial neural networks. (2008).
44. Ripley, B., Venables, W., & Ripley, M. B. Package 'nnet'. R package version. **7**, 3–12 (2016).
45. Yao, Z., & Ruzzo, W. L., (eds). A regression-based K nearest neighbor algorithm for gene function prediction from heterogeneous data. BMC bioinformatics. *BioMed. Central* (2006).
46. Cunningham, P., & Delany, S. J. k-Nearest Neighbour Classifiers. arXiv preprint http://arxiv.org/abs/200404523. (2020).
47. Kuhn, M., Wing, J., Weston, S., Williams, A., Keefer, C., Engelhardt, A. *et al*. Package 'caret'. The R Journal. (2020).
48. Sutton, C. D. Classification and regression trees, bagging, and boosting. *Handbook Stat.* **24**, 303–329 (2005).
49. Bengio, Y. *Learning Deep Architectures for AI* (Now Publishers Inc, 2009).
50. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **61**, 85–117 (2015).
51. Elbashir, M. K., Ezz, M., Mohammed, M. & Saloum, S. S. Lightweight convolutional neural network for breast cancer classification using RNA-seq gene expression data. *IEEE Access* **7**, 185338–185348 (2019).
52. Ciregan, D., Meier, U., & Schmidhuber, J. (eds.) Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition,* IEEE, (2012).
53. Mostavi, M., Chiu, Y.-C., Huang, Y. & Chen, Y. Convolutional neural network models for cancer type prediction based on gene expression. *BMC Med. Genom.* **13**, 1–13 (2020).
54. Friedman, J., Hastie, T. & Tibshirani, R. *The Elements of Statistical Learning* (Springer, 2001).
55. Yang, Z., Yu, Y., You, C., Steinhardt, J., Ma, Y. (eds). Rethinking bias-variance trade-off for generalization of neural networks. In *International Conference on Machine Learning,* PMLR, (2020).
56. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Adv. Neural. Inf. Process. Syst.* **25**, 1097–1105 (2012).
57. Simonyan, K., Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv preprint http://arxiv.org/abs/14091556. (2014).
58. Wei, R., Wang, J., Jia, W., & Wei, M. R. Package 'multiROC'. Technical report, CRAN, June 26, (2018).
59. Xiao, J. *et al*. Identification of microRNA precursors based on random forest with network-level representation method of stem-loop structure. *BMC Bioinform.* **12**(1), 165 (2011).
60. Batuwita, R. & Palade, V. microPred: effective classification of pre-miRNAs for human miRNA gene prediction. *Bioinformatics* **25**(8), 989–995 (2009).
61. Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002).
62. Chawla, N. V. *Data mining for imbalanced datasets: An overview* 875–886 (Springer, 2009).
63. Johnson, J. M. & Khoshgoftaar, T. M. Survey on deep learning with class imbalance. *J. Big Data* **6**(1), 27 (2019).
64. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H. & Herrera, F. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **42**(4), 463–484 (2011).
65. Blagus, R., & Lusa, L. SMOTE for high-dimensional class-imbalanced data. BMC Bioinform. **14**, 106. Epub 2013/03/26. https://doi.org/10.1186/1471-2105-14-106. PubMed PMID: 23522326; PubMed Central PMCID: PMCPMC3648438, (2013).
66. Trawiński, B., Smętek, M., Telec, Z. & Lasota, T. Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. *Int. J. Appl. Math. Comput. Sci.* **22**(4), 867–881 (2012).
67. Wang, H. *et al*. Comparison of machine learning methods for classifying mediastinal lymph node metastasis of non-small cell lung cancer from 18 F-FDG PET/CT images. *EJNMMI Res.* **7**(1), 1–11 (2017).
68. Ogutu, J. O., Schulz-Streeck, T., & Piepho, H. -P., (eds.) Genomic selection using regularized linear regression models: ridge regression, lasso, elastic net and their extensions. In *BMC Proceedings*, Springer, (2012).
69. Van Hulse, J., Khoshgoftaar, T. M., & Napolitano, A. (eds.) Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th International Conference on Machine Learning*, (2007).

Author contributions

All authors contributed substantially to this work. M.M. computed the features, generated the prediction model, performed experimental comparison and drafted the manuscript. H.M., I.B.M., M.K.E., and B.O. participated in the design of the study and helped to draft the manuscript. All authors reviewed the drafts of this manuscript and approved the final version for submission.

Funding

This work was funded by GSK Africa Non-Communicable Disease Open Lab through the DELTAS Africa Sub-Saharan African Consortium for Advanced Biostatistics (SSACAB) Grant No. 107754/Z/15/Z-training programme. The views expressed in this publication are those of the author(s) and not necessarily those of GSK.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-021-95128-x.

Correspondence and requests for materials should be addressed to M.M.

Reprints and permissions information is available at www.nature.com/reprints.

PLOS ONE

RESEARCH ARTICLE

Predictors of colorectal cancer survival using cox regression and random survival forests models based on gene expression data

Mohanad Mohammed[1,2]*, **Innocent B. Mboya**[1,3], **Henry Mwambi**[1], **Murtada K. Elbashir**[4], **Bernard Omolo**[1,5,6]

1 School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Pietermaritzburg, Scottsville, South Africa, **2** Faculty of Mathematical and Computer Sciences, University of Gezira, Wad Madani, Sudan, **3** Department of Epidemiology and Biostatistics, Kilimanjaro Christian Medical University College (KCMUCo), Moshi, Tanzania, **4** College of Computer and Information Sciences, Jouf University, Sakaka, Saudi Arabia, **5** Division of Mathematics & Computer Science, University of South Carolina-Upstate, Spartanburg, United States of America, **6** School of Public Health, Faculty of Health Sciences, University of Witwatersrand, Johannesburg, South Africa

* mohanadadam32@gmail.com

Citation: Mohammed M, Mboya IB, Mwambi H, Elbashir MK, Omolo B (2021) Predictors of colorectal cancer survival using cox regression and random survival forests models based on gene expression data. PLoS ONE 16(12): e0261625. https://doi.org/10.1371/journal.pone.0261625

Editor: Afnizanfaizal Abdullah, University of Technology Malaysia: Universiti Teknologi Malaysia, MALAYSIA

Received: July 15, 2021

Accepted: December 6, 2021

Published: December 29, 2021

Data Availability Statement: The dataset is publicly available on the Gene Expression Omnibus (GEO) public database (https://www.ncbi.nlm.nih.gov/geo/) under the accession number GSE39582.

Funding: This work was funded by GSK Africa Non-Communicable Disease Open Lab through the DELTAS Africa Sub-Saharan African Consortium for Advanced Biostatistics (SSACAB) Grant No. 107754/Z/15/Z- training programme. The views expressed in this publication are those of the

Abstract

Understanding and identifying the markers and clinical information that are associated with colorectal cancer (CRC) patient survival is needed for early detection and diagnosis. In this work, we aimed to build a simple model using Cox proportional hazards (PH) and random survival forest (RSF) and find a robust signature for predicting CRC overall survival. We used stepwise regression to develop Cox PH model to analyse 54 common differentially expressed genes from three mutations. RSF is applied using log-rank and log-rank-score based on 5000 survival trees, and therefore, variables important obtained to find the genes that are most influential for CRC survival. We compared the predictive performance of the Cox PH model and RSF for early CRC detection and diagnosis. The results indicate that *SLC9A8*, *IER5*, *ARSJ*, *ANKRD27*, and *PIPOX* genes were significantly associated with the CRC overall survival. In addition, age, sex, and stages are also affecting the CRC overall survival. The RSF model using log-rank is better than log-rank-score, while log-rank-score needed more trees to stabilize. Overall, the imputation of missing values enhanced the model's predictive performance. In addition, Cox PH predictive performance was better than RSF.

Introduction

Colorectal cancer (CRC) is the second leading cause of mortality in women and third in men [1]. The American cancer society estimate, about 1 in 23 men and 1 in 25 women develop colorectal cancer in their lifetime [2]. Globally, there were about 19.3 million new cancer cases in 2020 alone, while close to 10 million deaths were recorded due to cancer [3]. CRC represents 9.4% of cancer deaths and 10% of newly diagnosed cancer cases [3]. The incidence and mortality in males are 10.6% and 9.3%, respectively, while the incidence and mortality in females are 9.4% and 9.5%, respectively [3]. Early detection of CRC can reduce mortality due improved

author(s) and not necessarily those of GSK. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

chemotherapy regimens and surgical techniques [4–6]. The prognosis and survival of early intervention with CRC patients are linked with tumor staging, where early diagnosis of the tumor is more likely to be curable [7]. The 5-year relative survival rates for patients with localized CRC was 91% in the USA between 2010 and 2016 [8]. However, the 5-year relative survival rates of CRC cases at regional and distant stages are 72% and 14%, respectively [8]. The main characteristics of the CRC are that it has high inter-patient and intra-tumor heterogeneity. Other factors such as environment, lifestyle, and diet can lead to further heterogeneity in the CRC occurrence and progression [9–11]. This heterogeneity leads to variations in response to treatment between individuals. Determining the molecular markers is clinically essential to help detect and precisely predict the prognosis of patients with CRC.

Researchers have developed many methods to determine the prognostic molecular markers to early detect and predict the prognosis of patients with CRC. These methods include univariate and multivariate Cox proportional hazard models, elastic net estimation, and random forests for survival prediction [4, 7, 12–15]. Previous studies such as, Abdul Aziz *et al.* [12] analyzed the CRC death using the Cox proportional hazard model, and they reported a 19 gene signature that could predict the survival of CRC patients with Dukes' B and C stages. In their work, Abdul Aziz *et al.* used SAM, *limma*, and t-test to identify the most significant genes based on microarray gene expression data. Dai *et al.* [4] conducted a survival analysis using univariate and multivariate Cox models based on three microarray datasets from GEO and one dataset from the TCGA database. They used the DEGs from each of the three microarray datasets, and they identified 105 mutual DEGs based on the intersection of the three DEGs lists. They conducted a protein-protein interaction network (PPI) of the DEGs, and they identified hub genes. To investigate the 44 hub genes' prognostic values in CRC, they conducted a survival analysis using the sample splitting and Cox regression models based on the TCGA dataset. Their results showed that two down-regulated and two up-regulated hub genes were significantly associated with the CRC patients' overall survival.

Bian *et al.* [7] analyzed data from four microarray datasets and identified DEGs from each of them. They identified the common genes across the four datasets, and this way, they obtained 53 genes. Then they utilized PPI, which identified ten essential genes according to their degree value, betweenness centrality, and closeness centrality. They used gene expression profiling interactive analysis (GEPIA) to apply survival analysis using the log-rank test based on the expression levels. Their results showed that four low expressed genes out of the ten genes were significantly related to unfavorable prognosis in the patients with CRC. Martinez-Romero *et al.* [14] identified a new set of gene markers associated with CRC to predict tumor progression and evolution towards inferior survival stages based on an integrated gene expression dataset of 1273 CRC samples. They compared the early and late stages of CRC using *limma* to identify the genes (2707 DEGs) that had a significant effect on CRC tumor progression. Then, they applied Kaplan-Meier to rank the genes based on the non-parametric log-rank test. Their results identified 429 essential genes in which overexpression is related to low survival rate and 336 crucial genes in which repression is associated with inferior survival. They validated the top 5 genes using an external cohort study and presented a good separation of the CRC samples into two low and high-risk groups.

A study by Pan *et al.* [13] proposed a predictive model based on RNASeq gene expression data. Their model uses the differentially expressed genes (DEGs) profiles. These profiles were obtained using the univariates and multivariate Cox regression, which was used to compare TNM stages to assess their predictive survival accuracy. Their results showed that 10 DEGs had a significant effect on CRC survival. Yan *et al.* [15] implemented random forests to identify biomarkers associated with survival in CRC based on a set of oligonucleotide microarray data. Their results showed that four genes had the potential to predict CRC survival.

To the best of our knowledge, RSF has not been used with gene expression data in the previous studies to predict CRC survival. The gene expression data is characterized by the problem of the curse of dimensionality and collinearity. To overcome this problem, the CRC survival is predicted based on selecting the differentially expressed genes (DEGs) in colorectal cancer that was based on the three-mutation status (KRAS, BRAF, and TP53) where they serve as a predictive biomarker of response to treatment in CRC. We assume that complex interaction between multiple DEGs contributes to prognostic survival differences between wild-type and mutant patients with CRC.

We developed and compared Cox proportional hazard (Cox PH) model and random survival forests (RSF) in predicting CRC survival and associated biomarkers using a public genome database from Gene Expression Omnibus (GEO). The aim was to assess the CRC survival predictors accounting for missing data based on the gene expression data. We selected 54 common differentially expressed genes from three mutations (KRAS, BRAF, and TP53), using the complete case samples, and performed analysis using Cox PH and RSF models before and after imputation.

Materials and methods

Dataset

The dataset with accession number GSE39582 [16], was downloaded from Gene Expression Omnibus (GEO) public database (https://www.ncbi.nlm.nih.gov/geo/) using the BRB-ArrayTools software (https://brb.nci.nih.gov/BRB-ArrayTools/). This dataset has 54675 probes taken from 566 samples with colon cancer and 19 non-tumor samples. Usually, the gene expression data includes noisy and or irrelevant genes. Therefore, performing data cleaning and feature (genes) selection are essential steps that should be applied before modeling the data. A pre-processing step was applied to prepare the dataset for modeling. These pre-processing steps are log2 transformation, quantile normalization, gene filtration, and differentially expressed genes analysis using a two samples t-test. Filtration is a process in data cleaning used to eliminate insufficiently expressed probes and those with excessive missing expression levels across the samples [17–20]. On the other hand, quantile normalization and log2-transformed steps to eliminate the variation between samples. BRB-ArrayTools is used to implement the filtration and normalization of the dataset. The two-sample t-test, with the 0.001 significance level threshold, was used for gene selection to provide informative genes for building survival models. The overall procedures that we followed in our analysis are summarized in Fig **1**.

Statistical analysis

We analyzed the gene expression data using the *R* version (R-4.0.4). Summary statistics of the gene expressions are depicted in the supplementary file (see S1 Appendix). These statistics include the minimum, maximum, means, and standard deviations of the expression levels. We used frequency and percentages for the categorical data representing the clinical information, as shown in Table **1**. The statistical analysis was conducted in three phases; the first phase is the complete case analysis, followed by imputation of missing values in the outcome based on the covariates and an appropriate imputation model. Then we applied survival analysis on the complete case and imputed datasets. The survival analysis results on these two datasets were compared to evaluate the precision of estimates. Two separate models were fitted before and after imputations; the first is the Cox regression model, while the second is the random survival forests with log-rank and log-rank-score split rules. The missing values were assumed to be missing at random (MAR), where the probability of data being missing does not depend on

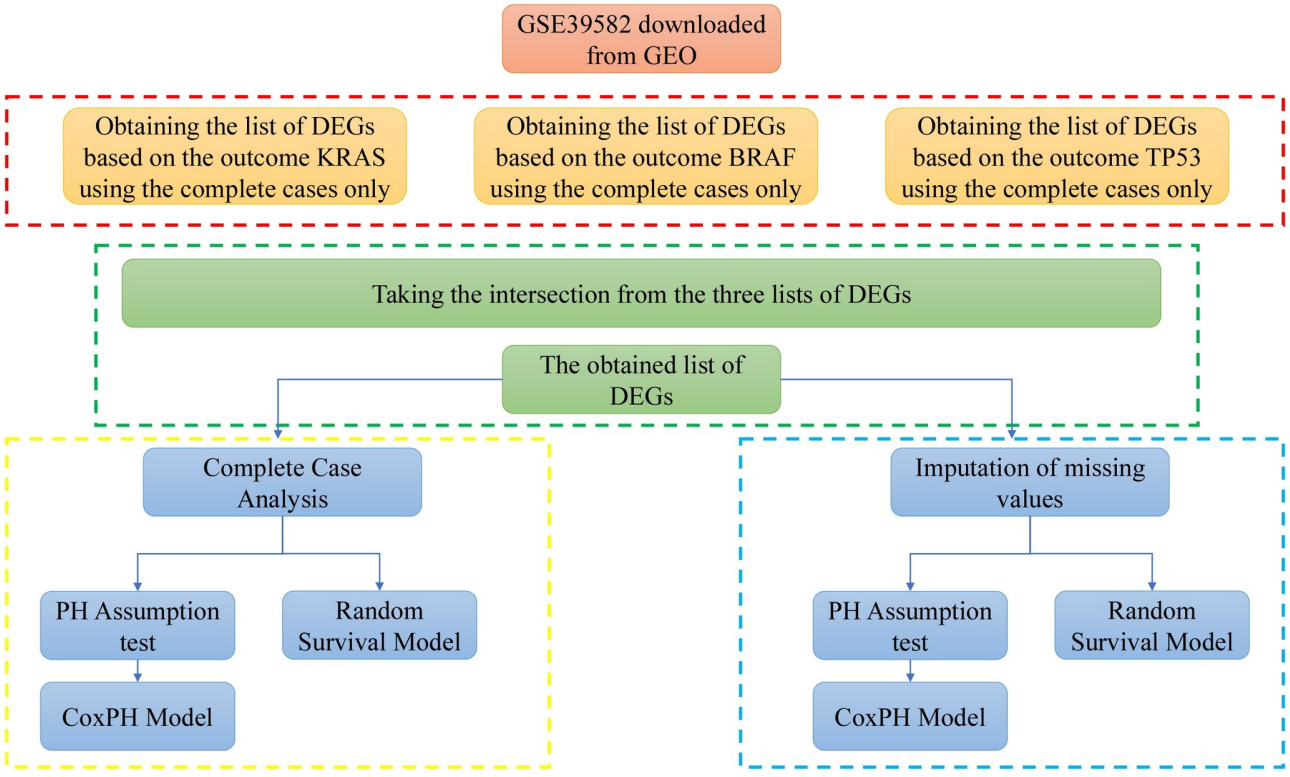


Fig 1. Flow-chart of the procedure followed in the pre-processing and analysis of the dataset.

https://doi.org/10.1371/journal.pone.0261625.g001

the unobserved data, conditional on the observed data [21–24]; consequently, the genes and other covariates in the dataset were used to predict missingness.

Complete case analysis. The filtration step resulted in 18865 out of 54675 probes. These 18865 probes were used for further reduction analysis using a t-test. To find the differentially expressed genes (DEGs) that discriminate between the mutant and wild-type mutation, we used the three mutation types, KRAS, BRAF, and TP53. We created three different datasets using the 18865 probes with each of the three mutation types based on these three mutation types. First, we removed the samples with missing values for each of the three datasets according to their clinical outcome. Then, we calculated the correlation matrix for the gene expression data and filtered out one gene from every two genes that show a correlation coefficient greater than 0.6. Subsequently, we extracted three DEGs lists from all three datasets using a two-sample t-test based on 0.001 thresholds. Ultimately, from the three lists of DEGs, there were 54 common genes (see S1 Appendix). Also, we used the common samples across the three datasets to produce the complete cases in one dataset. The samples with missing or zero values in the event status and time variables were removed. We then converted the five TNM stages into a new categorical variable with two stages (Early and Late), where stages four and five were combined to give the late category. Finally, we used the obtained data for finding the most significant gene markers that may predict survival for CRC patients. **Table 2** provides a concise summary of the pre-processed data.

Multiple imputations of the missing values. To compensate for the missing data, we used the R package "mice (Multivariate Imputation by Chained Equations)", which impute the missing values in the covariates. The mice package takes care of uncertainty related to missing

Table 1. Clinical characteristics of colorectal cancer patients (N = 307).

Variable	Frequency (n)	Percentage (%)
Age at diagnosis in years: Mean (SD*)	66.8 (13.2)	
KRAS Mutation		
Mutant	123	40
WildType	184	60
BRAF Mutation		
Mutant	25	8
WildType	282	92
TP53 Mutation		
Mutant	166	54
WildType	141	46
Tumor Location		
Proximal	124	40
Distal	183	60
Cancer stage		
Early	156	51
Late	151	49
Sex		
Female	137	45
Male	170	55
Molecular subtype		
C1	65	21
C2	49	16
C3	43	14
C4	29	9
C5	29	9
C6	36	12

*SD: Standard deviation

https://doi.org/10.1371/journal.pone.0261625.t001

values [23–25]. It assumes that the missing values are missing at random (MAR) see (Fig 2), where the probability of missing data does not depend on the unobserved data, conditional on the observed data [21–24]. The mice package uses the genes and other covariates in the dataset to predict missingness. The missingness pattern in the data is assumed to be non-monotone. In this pattern, some subject values can be observed again after missing values happen [23–25]. For this missing data pattern, it is recommended to use the chained equations (fully conditional specification (FCS)) [26], or the Markov Chain Monte Carlo (MCMC) method to impute missing values [25].

Table 2. Summary of the filtered datasets and the pre-processing steps.

Dataset (GSE39582) *		Number of samples	Complete cases	Common samples	Total number of genes	After filtration	Uncorrelated genes	DEGs (t-test)	Common genes
Clinical outcomes	KRAS	585	545	307	54675	18865	13827	711	54
	BRAF		512					2388	
	TP53		351					629	

* Three datasets with the same covariates and different clinical outcome

https://doi.org/10.1371/journal.pone.0261625.t002

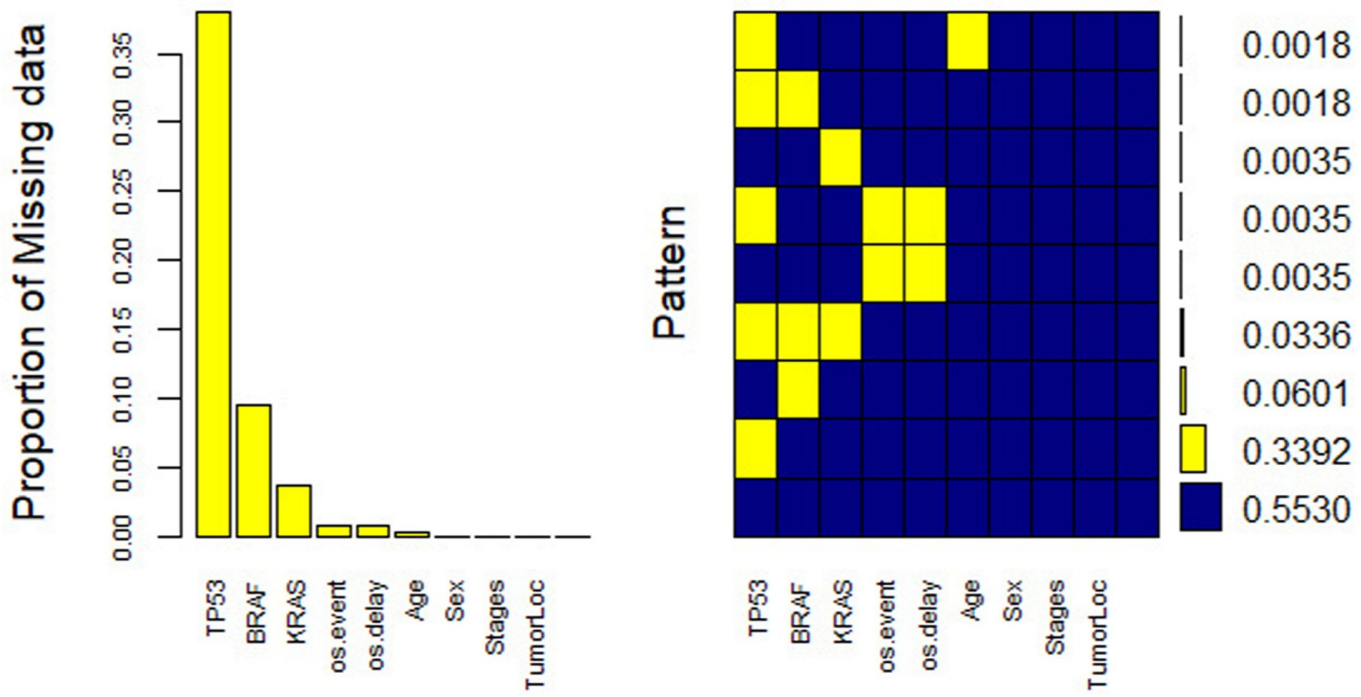


Fig 2. Proportion and patterns of missing values in the clinical characteristics available in the GSE39582 dataset.

https://doi.org/10.1371/journal.pone.0261625.g002

We used FCS to handle the missing values in our dataset implemented in the mice package in *R* using a random forest model. The FCS is considered a powerful and statistically valid method for creating imputations in both categorical and continuous variables [26]. We generated 5 imputed datasets using random forest (rf) imputations after 100 iterations (imputation cycles). We used 1051991 as a random seed to replicate imputation results each time a multiple imputation analysis was performed. In addition, we followed the procedures indicated by the work of Sterne *et al.* [27] for reporting and analysis of missing data. KRAS, BRAF, TP53, and the event status were imputed as binary, while time and age imputed as numeric variables. The rest of the variables did not contain any missing values, and were used as auxiliary variables in the imputation model. Overall, firstly we performed a complete case analysis using Cox PH and random survival forests models. Thereafter, we compared the final models from this analysis to those from the multiply imputed dataset.

Experimental setup. To evaluate the different methods, the resulting dataset was divided into training set (80%) and testing set (20%). The training set was then divided into 10 subsets to train the methods using 10-fold cross validation approach to avoid overfitting. In the 10-fold cross-validation approach the integrated brier scores (IBS) is calculated on each fold left-out while the model is trained on the other 9 folds. Finally, the trained model is tested on the testing set. The model performance was measured using prediction error curve (pec).

Statistical methods

Cox proportional hazard model (Cox PH). Cox proportional hazard model is the most widely used statistical model for modeling time to event data [28]. The Cox PH evaluates the association of the survival time of patients and one or more predictors/genes variables. The Cox PH model relates the effect of predictors which include genes in our case to the rate or

hazard of occurrence of an event such time to infection, death, recurrence of a condition at a certain point of time, this rate is generally referred as the hazard rate [29, 30]. In order to estimate the association of the gene expression levels and the survival time, consider n cancer samples say from sample $i = 1,2,\ldots,n$ and $\mathbf{g}_i = (g_{i1}, g_{i2}, g_{i3},\ldots,g_{ip})$ is a vector of p genes expression level. The i^{th} patient survival data can be represented by $(T_i, \delta_i, g_{i1}, g_{i2}, g_{i3},\ldots,g_{ip})$, where $i = 1,2,\ldots,n$; T_i and δ_i indicate the survival time and the censor status respectively. The Cox PH model is mathematically represented as follow

$$h_i(t) = h_0(t)e^{\boldsymbol{\beta}' \boldsymbol{g}_i} \tag{1}$$

where the parameters vector $\boldsymbol{\beta}'$ is the regression coefficients and $\boldsymbol{g}_i$ is the covariates (genes) vector. The baseline hazard function $h_0(t)$ is unspecified and non-parametric function of an individual with all expression levels equal to zero [12, 31]. The model has a parametric part specified by the linear predictor and assumed to be proportional to the non-parametric baseline hazard. This means that for two individuals, i and j, the hazard ratio is

$$\frac{h_i(t)}{h_j(t)} = \frac{e^{\boldsymbol{\beta}' \boldsymbol{g}_i}}{e^{\boldsymbol{\beta}' \boldsymbol{g}_j}} \tag{2}$$

The hazard ratio is assumed to be independent of time t. The maximum partial likelihood method used to estimate the Cox PH model parameters is given by

$$L(\beta) = \prod_{r \in E} \frac{e^{\boldsymbol{\beta}^T \boldsymbol{g}_r}}{\sum_{j \in R_r} e^{\boldsymbol{\beta}^T \boldsymbol{g}_j}} \tag{3}$$

where E indicates the indices of the events (e.g., deaths) and R_r represents the vector of indices of the individuals at risk at time t_r - 0. The results of the Cox PH model are easy to interpret, however, there are key assumptions needed such as linearity and proportional hazards. We used *survival* and *survminer* packages to implement Cox PH model in *R*.

Moreover, we performed the stepwise regression for developing the Cox PH model at a 5% threshold level to find a simple model that shows the essential genes (markers) and clinical covariates correlated with the CRC. At each time, we removed the genes/ covariates that are not significant at $\alpha = 0.05$ level of significance. Thereafter, we tested for the Cox PH assumption, and the integrative analysis of the CRC data showed five genes (markers) that passed the Cox PH assumption test. Thereafter, we used the five genes and the other clinical information to fit the Cox PH model.

Random survival forests (RSF). Random survival forests are an ensemble of trees and a non-parametric method constructed by bagging of classification trees for right censored data [32, 33]. The RSF are an extension of the random forests method proposed by Breiman [34]. It works on high dimensional data where the number of covariates exceeds the number of the observations. Also it can handle data that consist of complex and non-linear relationships between the dependant and the independent variables and when the covariates violate the proportional hazard assumption [35]. There are several advantageous of using the RSF method, such as, it is not based on any model assumption compared to Cox PH model. It seeks to find a model that best represent the data in the case of limited survival data. In addition, it can handle high dimensional data unlike Cox PH, and it is robust to outliers in the explanatory variables [33]. RSF employs two steps of randomizations to grow the tree. These two steps are the bootstrap sample to select cases randomly and random selection of subset of covariates for splitting the nodes of the tree. These two steps help to decorrelate the tree [20, 33]. The RSF was implemented using the *randomForestSRC* package in *R* [36].

Random survival forests algorithm. We used the RSF algorithm that was introduced in the work of Ishwaran *et al.* [32] as shown below:

For *i* in 1: *ntrees*

- Draw bootstrap samples from the original total number of samples. For each bootstrap exclude approximately 37% of the samples as out-of-bag (OOB) samples.
- Build a survival tree for every bootstrap sample by recursively repeating the following steps for each node in a tree

○ Randomly select v genes at random from the p genes ($v = \sqrt{p}$)

○ To split the node, pick the best gene among the v genes, that maximizes survival differences between daughter nodes. We used log-rank and log-rank-score splitting rules as measures of survival differences.

○ Produce the tree to full size under the constraint that a terminal node should have no less than $d_0 > 0$ unique deaths.

○ Calculate a cumulative hazard function (CHF) for every tree. Average the CHF for all the *ntrees* trees to find the ensemble CHF.

○ Calculate the OOB prediction error for the ensemble CHF, using OOB samples.

Once the survival tree is built, the ends of the tree are called the terminal nodes. Assume, the terminal node is h and $t_{n,h}$ is the individual's death time at node h, $d_{n,h}$ is the number of deaths, and $M_{n,h}$ is the number of individuals at risk at time $t_{n,h}$. Therefore, the cumulative hazard function (CHF) can be estimated using the Nelson-Aalen estimator [37] as follows

$$\hat{H}_h(t) = \sum_{t_{n,h} \le t} \frac{d_{n,h}}{M_{n,h}} \tag{4}$$

The CHF was calculated for all the terminal nodes. The CHF for new observation i given a vector of genes as a covariate $\mathbf{g}_i$, can be calculated for one tree as follows

$$\hat{H}_h(t|\mathbf{g}_i) = \hat{H}_h(t), \qquad \text{for } \mathbf{g}_i \in h \tag{5}$$

To compute an ensemble CHF, the average of the *ntrees* trees is calculated, and the bootstrap ensemble CHF for an observation i is

$$\hat{H}_e(t|\mathbf{g}_i) = \frac{1}{ntrees} \sum_{b=1}^{ntrees} \hat{H}_b(t|\mathbf{g}_i) \tag{6}$$

let,

$$I_{i,b} = \begin{cases} 1 & \text{if } i \text{ is an OBB observation for } ntrees \text{ training sample.} \\ 0 & \text{Otherwise.} \end{cases} \tag{7}$$

then the OOB ensemble CHF for an observation i is given by

$$\hat{H}^*_e(t|\mathbf{g}_i) = \frac{\sum_{b=1}^{ntrees} I_{i,b} \hat{H}^*_b(t|\mathbf{g}_i)}{\sum_{b=1}^{ntrees} I_{i,b}} \tag{8}$$

therefore, $\hat{H}^*_e(t|\mathbf{g}_i)$ is an average over the training samples where i is an OOB observation.

Log-rank split rule. The log-rank split-rule is a measure of a node separation which helps in determining the best split for that node [38]. Let h be a node of a tree and let there are n

individuals with this node. Suppose (T_1, σ_1), (T_2, σ_2), . . ., (T_n, σ_n) are the survival outcomes corresponding to the n individuals. Thus, the best split at node h on covariate x at split point c, is the one that maximize the log-rank statistic between the two daughter nodes [32] given as follow

$$L(x,c) = \frac{\sum_{i=1}^{N}(d_{i1} - Y_{i1}\frac{d_i}{Y_i})}{\sqrt{\sum_{i=1}^{N}\frac{Y_{i1}}{Y_i}(1-\frac{Y_{i1}}{Y_i})(\frac{Y_i-d_i}{Y_i-1})d_i}} \tag{9}$$

The aim is to maximize the log-rank statistic by finding values of x and c that maximize $L(x, c)$. Specifically, we are looking to find a predictor x^* and c^* such that $|L(x^*, c^*)| \geq |L(x, c)|$ for every x and c. This process is repeated at every node until the terminal node is reach.

Log-rank-score split rule. The log-rank-score split rule is a version of the log-rank-score split rule [39]. Consider $r = (r_1, r_2, \ldots, r_n)$ as a vector that ranks the survival times $(T, \delta) = ((T_1, \sigma_1), (T_2, \sigma_2), \ldots, (T_n, \delta_n))$ [39, 40]. Assume $a = a(T, \delta) = (a_1(r), a_2(r), \ldots, a_n(r))$ indicates the ranked score vector. Let the ranked vector r order the genes variables in such a way that $g_1 < g_2 < \cdots < g_n$. Therefore, the log rank score for an observation at T_i is given by

$$a_i = a_i(T,\delta) = \delta_i - \sum_{j=1}^{\gamma_i(T)} \frac{\delta_j}{(n - \gamma_j(T) + 1)}, \tag{10}$$

where, $\gamma_j(T) = \sum_{i=1}^{n} \chi\{T_i \leq T_j\}$ is the number of individuals who died or were censored before or at time T_j.

Performance evaluation

We used integrated brier scores (IBS) measure [41] to assess and compare the accuracy of the predictive performance of all the models in this study. The IBS represent the average squared differences between the observed survival status and the predicted survival probability at time t. However, the value of the IBS is always between 0 and 1, the value of 0 represent the best possible IBS value. We calculated the brier scores (BS) measure using the test sample of size n_{test} as follows

$$BS(t) = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \left\{ [0 - \hat{S}(t|x)]^2 \frac{I(t_i \leq t, \delta_i = 1)}{\hat{G}(t_i|x)} + [1 - \hat{S}(t|x)]^2 \frac{I(t_i > t)}{\hat{G}(t|x)} \right\} \tag{11}$$

where $\hat{G}(t|x) \approx P(C > t|X = x)$ is the Kaplan-Meier estimate for the conditional survival function of the censoring times. Therefore, the IBS is calculated as below

$$IBS = \int_0^{\max(t)} BS(t)\, dt \tag{12}$$

Results

Cox proportional hazards analysis

The results of the survival problem based on gene expression data were obtained using *R*. We used the Cox PH model based on the selected covariates that satisfy the Cox PH assumptions. We tested the Cox PH assumptions using the Schoenfeld residual test implemented by the function *cox.zhp*. The Cox PH model assumes the regression parameters are constant over time. Therefore, the hazard ratios for any two individuals are constant over time. However, the covariates that do not satisfy the Cox PH assumptions do not meet the criteria to be entered in our final Cox PH model. As a first step, we fitted the Cox PH model for all the covariates

Table 3. Testing the proportional hazard assumption using scaled Schoenfeld residuals.

Probeset ID (Symbol)	χ^{2*} (df)	p-value
204014_at (DUSP4)	10.219 (1)	0.0014
212947_at (SLC9A8)	1.345 (1)	0.2462
218611_at (IER5)	2.045 (1)	0.1527
219973_at (ARSJ)	3.601 (1)	0.0577
221522_at (ANKRD27)	1.583 (1)	0.2083
221605_s_at (PIPOX)	1.651 (1)	0.1988
227134_at (SYTL1)	4.699 (1)	0.0302
Age at diagnosis (years)	2.589 (1)	0.1076
Molecular subtype	15.824 (5)	0.0074
Disease stages	1.173 (1)	0.2787
Sex	0.378 (1)	0.5388
Tumor location	0.951 (1)	0.3294

*Chi-square statistic

https://doi.org/10.1371/journal.pone.0261625.t003

(genes and clinical variables) in our dataset and then obtained the Cox PH assumption using the Schoenfeld residuals Table 3. The genes and variables in violation of the Cox PH assumption ($p<0.05$) were DUSP4, SYTL1, and molecular subtype.

From the Cox PH model in Table 3, three variables violated the Cox PH assumption, and therefore, these genes and molecular subtype were not included in the final Cox PH model. We fitted the Cox PH model on the genes and variables that did not violate the Cox PH assumptions before and after imputation. The results from this analysis are shown in Table 4. Results before imputation of missing values indicated that *218611_at (IER5)* (HR = 9.51, 95% CI 1.30, 69.58), 221522_at (*ANKRD27*) (HR = 34.89, 95%CI 1.91, 635.90), and late disease

Table 4. Multivariable Cox PH results for predictors of colorectal cancer survival among adults aged 24 years and above.

Probeset ID (Symbol) / Variables	Before imputation (N = 307)			After imputation (N = 566)		
	HR* (SE)	95%CI	P-value	HR* (SE)	95%CI	P-value
212947_at (SLC9A8)	0.09 (0.84)	(0.02, 0.49)	0.005**	0.30 (0.66)	(0.08, 1.07)	0.066
218611_at (IER5)	9.51 (1.02)	(1.30, 69.58)	0.027*	6.48 (0.79)	(1.37, 30.53)	0.019*
219973_at (ARSJ)	0.23 (0.48)	(0.09, 0.58)	0.002**	0.44 (0.36)	(0.22, 0.89)	0.024*
221522_at (ANKRD27)	34.89 (1.48)	(1.91, 635.90)	0.016*	2.49 (1.06)	(0.31, 19.95)	0.393
221605_s_at (PIPOX)	0.43 (0.34)	(0.22, 0.85)	0.014*	0.49 (0.27)	(0.28, 0.83)	0.009**
Age diagnosis (years)	1.03 (0.01)	(1.01, 1.05)	0.001***	1.03 (0.01)	(1.01, 1.04)	<0.000***
Sex						
Female	1.00			1.00		
Male	1.23 (0.20)	(0.84, 1.81)	0.281	1.40 (0.15)	(1.05, 1.88)	0.024
Stages						
Early	1.00			1.00		
Late	1.97 (0.20)	(1.33, 2.93)	0.001***	1.96 (0.15)	(1.47, 2.63)	<0.000***
Tumor location						
Proximal	1.00			1.00		
Distal	1.06 (0.21)	(0.71, 1.58)	0.783	0.86 (0.16)	(0.63, 1.18)	0.356

HR: Hazard ratio, SE: Standard error, adjusted for 212947_at, 218611_at, 219973_at, 221522_at, 221605_s_at, age at first diagnosis, sex, disease stage, and tumor location.

https://doi.org/10.1371/journal.pone.0261625.t004

stage (HR = 1.97, 95%CI 1.33, 2.93) were associated with higher hazards of death. However, we note that two confidence intervals for *IER5* and *ANKRD27* are quite wide; therefore, they should be interpreted caution. For every year increase, the hazards of death increased by 1.03 (95%CI 1.01, 1.05). Significantly lower hazards were observed in *212947_at (SLC9A8)* (HR = 0.09, 95%CI 0.02, 0.49), *219973_at (ARSJ)* (HR = 0.23, 95%CI 0.09, 0.58), and 221605_s_at (*PIPOX*) (HR = 0.43, 95%CI 0.22, 0.85) differentially expressed genes.

After imputation of missing values, the Cox PH model showed that sex was a significant predictor of males having higher death hazards (HR = 1.40, 95%CI 1.05, 1.88) than females. Also, the disease stage covariate was a significant predictor where those with late disease stage had higher death hazards (HR = 1.96, 95%CI 1.47, 2.63) than early cases. Moreover, the results illustrated that *219973_at (ARSJ)* (HR = 0.44, 95%CI 0.22, 0.89), 221605_s_at (*PIPOX*) (HR = 0.49, 95%CI 0.28, 0.83) were related with lower hazards of death. For every year increase, the hazards of death increased by 1.03 (95%CI 1.01, 1.04). Significantly higher hazards were detected with gene *218611_at (IER5)* (HR = 6.48, 95%CI 1.37, 30.53) gene.

Random survival forests analysis

We fitted two random survival forests models, including survival trees built using log-rank and the log-rank-score split rules on the datasets before and after imputation. These two models were built using the 54 genes and the other clinical information as covariates. The characteristics of the two fitted models are summarized in Table **5** below.

Permutation importance measure used to identify the most important genes/ clinical variables associated with the survival of the colon patients [42–44]. We fitted a random survival forest model before imputation and after imputation with 5000 survival trees built using log-rank and log-rank-score and their results presented in Figs **3** and **4**.

Table **5** and Fig **3** show that the log-rank split-rule is more stable than the log-rank-score split-rule. Moreover, we fitted the model with 1000, 2000, and 3000 survival trees and noticed that the log-rank-score spilt-rule needs more survival trees to stabilize. In addition, the error rate for the forest built with survival trees based on the log-rank and log-rank-score split-rules are 41.26 and 49.05, respectively. These error rates of the RSF before imputation are much

Table 5. Random survival forests results before and after imputation using log-rank and log-rank-score split rules.

	Before imputation (N = 246)*		After imputation (N = 453)*	
	Log-rank	**Log-rank-score**	**Log-rank**	**Log-rank-score**
Number of deaths	88	88	157	157
Number of trees	5000	5000	5000	5000
Forest terminal node size	15	15	15	15
Average no. of terminal nodes	13.58	11.92	25.34	22.14
No. of variables tried at each split	8	8	8	8
Total no. of variables	62	62	62	62
Resampling used to grow trees	swor	swor	swor	swor
Resample size used to grow trees	155	155	286	286
Analysis	RSF	RSF	RSF	RSF
Family	surv	surv	surv	surv
Splitting rule	log-rank	log-rank-score	log-rank	log-rank-score
Number of random split points	10	10	10	10
Error rate	41.26%	49.05%	33.22%	43.01%

* Analysis performed using the 80% training set

https://doi.org/10.1371/journal.pone.0261625.t005

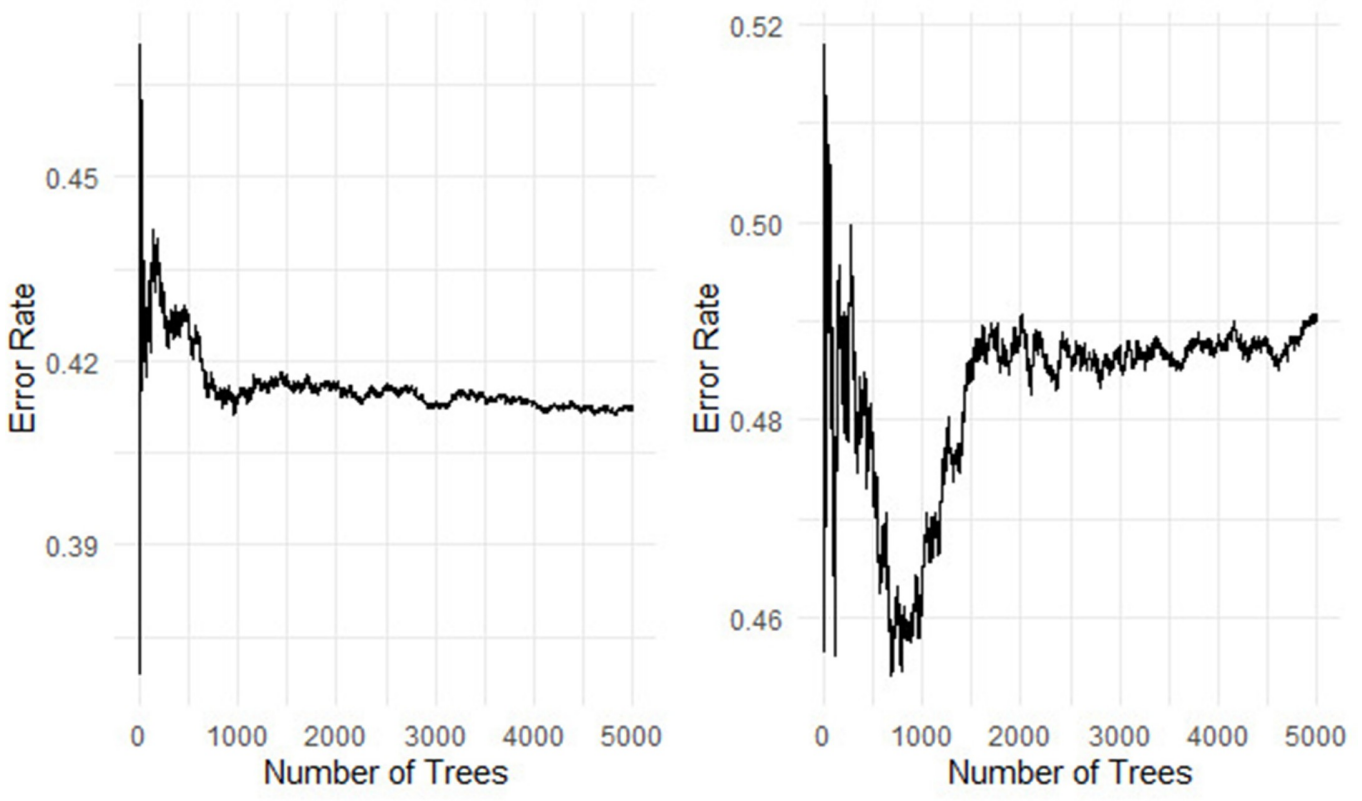


Fig 3. The prediction error rate for the random survival forests of 5000 trees before imputation and the log-rank and log-rank-score in the left and right panel used 80% training dataset.

https://doi.org/10.1371/journal.pone.0261625.g003

higher than the error rates for RSF built after imputation, as shown in Table 5. This result indicates that the imputation can improve the performance of RSF.

The genes/ covariates associated with CRC ranked using RSF according to their importance before and after imputation based on the log-rank, and log-rank-score split-rules are presented in Figs 5 and 6. Using RSF allows all 54 genes and other covariates regardless of their satisfying the Cox PH assumption. However, this is a very important characteristic of the RSF, as explained in the model building stage. The selection of the genes/ covariates in the model does not need to satisfy the too restrictive Cox PH assumption. RSF is purely non-parametric; hence there is no requirement of the Cox PH assumption being satisfied a prior.

We implemented RSF with 5000 survival trees built using two split-rules before and after imputation. The RSF identified the most important genes/ covariates that explain the survival of CRC patients by calculating the measure of the permutation importance as a variable's importance [32, 43]. For the RSF before imputation see (Fig 5), the top 20 genes/ covariates that are most important and strongly associated with the CRC obtained using the log-rank split-rule are age, *SLC28A3*, stages, *TNFSF9*, *EGLN3*, molecular subtype, *CTSV*, *ANKRD27*, *POLR3B*, *CTSA*, *SYTL1*, *MYRF*, *RPS27L*, *L3MBTL1*, *PIPOX*, *ADPRM*, *SLC6A4*, *LDLRAD3*, *MSRA*, and *SCAND1*. While the top 20 genes/ covariates that were identified by RSF using log-rank-score are *POLR3B*, *L3MBTL1*, *CTSV*, *EGLN3*, *SYTL1*, age, molecular subtype, *LDLRAD3*, *MAP7D2*, *SLC28A3*, *ANKRD27*, stages, *SLC6A4*, *CTSA*, *CABLES2*, *TNFSF9*, *GIF*, *SCAND1*, *PTP4A3*, and *MSRA*.

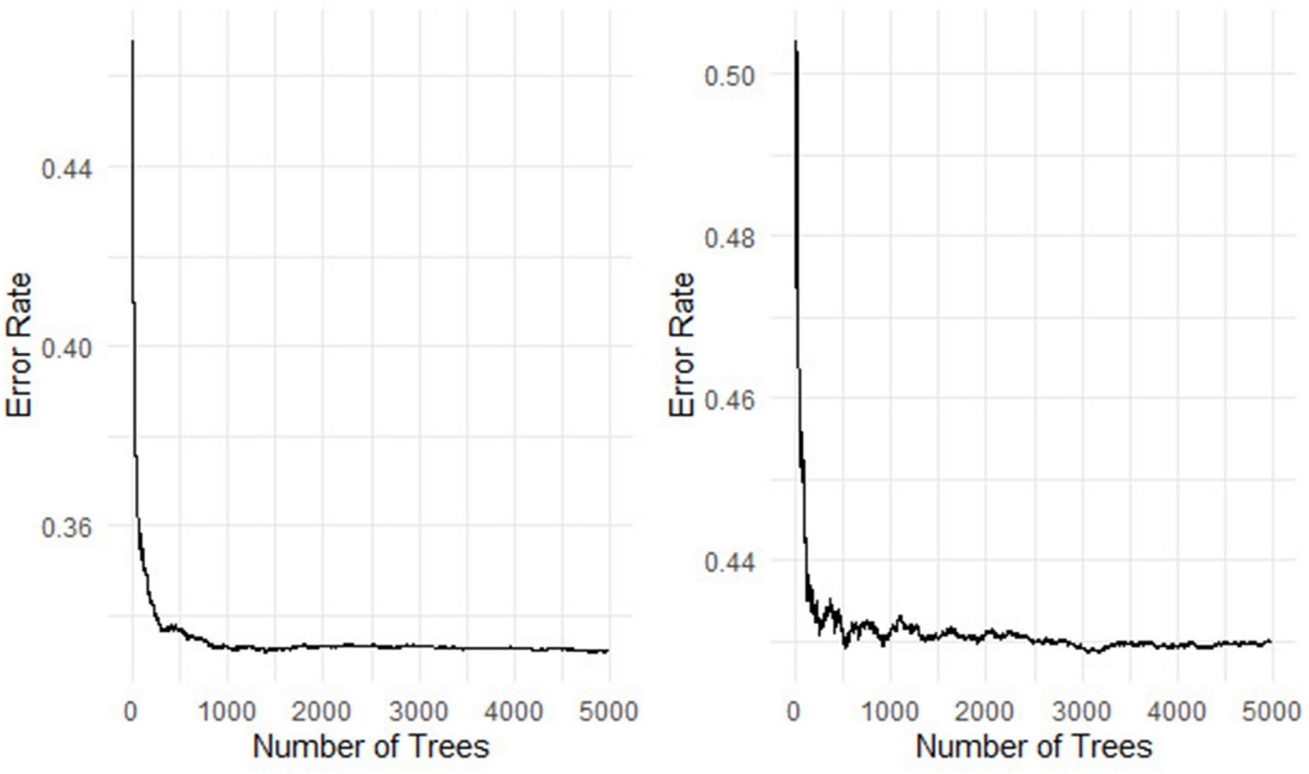


Fig 4. The prediction error rate for random survival forests of 5000 trees after imputation and the log-rank and log-rank-score in the left and right panel, respectively, using 80% training dataset.

https://doi.org/10.1371/journal.pone.0261625.g004

However, for the RSF after imputation (Fig 6), the top 20 genes/ covariates strongly related to CRC identified using RSF with log-rank split-rule are age, stages, molecular subtype, *PIPOX*, *ADPRM*, *CLCN2*, *RPS27L*, *IER5*, *POLR3B*, *SLC6A4*, *KRAS*, *SGMS2*, *DUSP4*, *SLC28A3*, *SLC9A8*, *ACOT8*, *SYTL1*, *CABLES2*, *SCAND1*, and *MAP7D2*. Although the RSF with log-rank-score obtains a top 20 genes/ covariates strongly relevant to CRC, these genes/ covariates are molecular subtypes, *POLR3B*, *CLCN2*, *IER5*, *SLC9A8*, *MAP7D2*, *CABLES2*, *SYTL1*, stages, *KRAS*, *SLC6A4*, *LDLRAD3*, *CTSA*, *SCAND1*, *PIPOX*, *ARSJ*, *PHACTR3*, *SLC28A3*, *SGMS2*, and *CTSV*.

The RSF with log-rank split-rule after imputation performed better in terms of the error rate. Age and disease stage were the most important covariates that affecting CRC. However, the *PIPOX*, *IER5*, and *SLC9A8* were among the most important genes strongly associated with CRC. These results agree with the results achieved from fitting the Cox PH model presented in Table 4. As far as significant effects are concerned, the most striking result to emerge was that the RSF model did pick other genes and covariates as substantial, e.g., molecular subtype and *DUSP4* which could not be included in the Cox PH model because of not satisfying the Cox PH assumption.

Predictive performance

We assessed the predictive performance of the models using the integrated brier scores measure in *R* using the *pec* package [45, 46]. The model with lower prediction error rates is therefore considered useful [43, 47]. Figs 7 and 8 show the prediction error curve of the RSF (log-

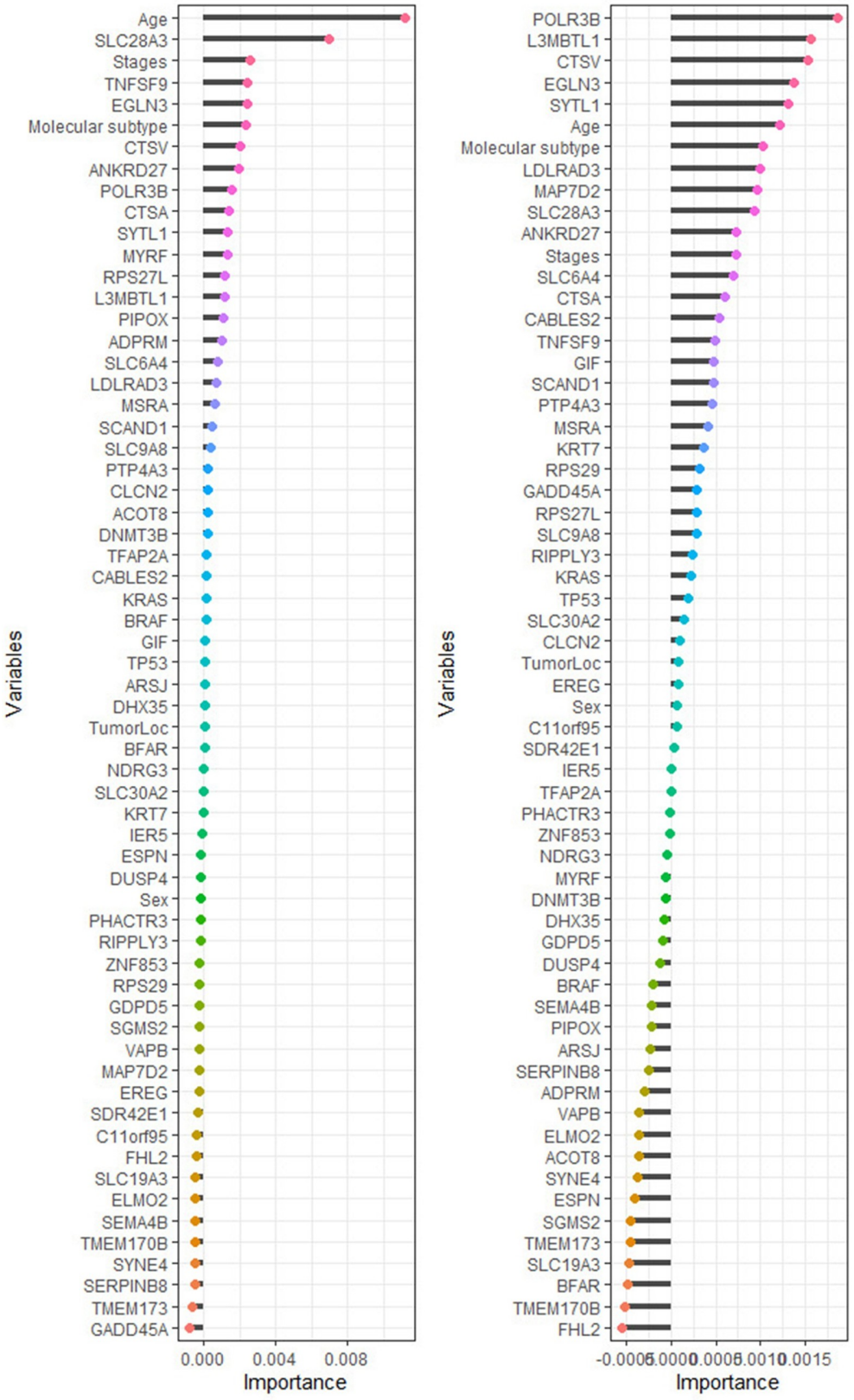

Age
SLC28A3
Stages
TNFSF9
EGLN3
Molecular subtype
CTSV
ANKRD27
POLR3B
CTSA
SYTL1
MYRF
RPS27L
L3MBTL1
PIPOX
ADPRM
SLC6A4
LDLRAD3
MSRA
SCAND1
SLC9A8
PTP4A3
CLCN2
ACOT8
DNMT3B
TFAP2A
CABLES2
KRAS
BRAF
GIF
TP53
ARSJ
DHX35
TumorLoc
BFAR
NDRG3
SLC30A2
KRT7
IER5
ESPN
DUSP4
Sex
PHACTR3
RIPPLY3
ZNF853
RPS29
GDPD5
SGMS2
VAPB
MAP7D2
EREG
SDR42E1
C11orf95
FHL2
SLC19A3
ELMO2
SEMA4B
TMEM170B
SYNE4
SERPINB8
TMEM173
GADD45A
Variables
0.000
0.004
0.008
Importance
POLR3B
L3MBTL1
CTSV
EGLN3
SYTL1
Age
Molecular subtype
LDLRAD3
MAP7D2
SLC28A3
ANKRD27
Stages
SLC6A4
CTSA
CABLES2
TNFSF9
GIF
SCAND1
PTP4A3
MSRA
KRT7
RPS29
GADD45A
RPS27L
SLC9A8
RIPPLY3
KRAS
TP53
SLC30A2
CLCN2
TumorLoc
EREG
Sex
C11orf95
SDR42E1
IER5
TFAP2A
PHACTR3
ZNF853
NDRG3
MYRF
DNMT3B
DHX35
GDPD5
DUSP4
BRAF
SEMA4B
PIPOX
ARSJ
SERPINB8
ADPRM
VAPB
ELMO2
ACOT8
SYNE4
ESPN
SGMS2
TMEM173
SLC19A3
BFAR
TMEM170B
FHL2
Variables
-0.0005
0.0000
0.0005
0.0010
0.0015
Importance

Fig 5. The rank of most predictive genes and clinical variables for colorectal cancer patients' survival before the imputation is based on how they influence the survival outcome. The variables importance is built using log-rank and log-rank-score split-rules in the left and right panel, respectively.

https://doi.org/10.1371/journal.pone.0261625.g005

rank and log-rank score) and Cox PH models before and after imputation. These prediction curves show that Cox PH outperformed RSF with log-rank and log-rank score split rules. The Cox PH model before and after imputation had similar prediction errors, while RSF models under the two split-rules (log-rank and log-rank-score, respectively) after imputation had lower prediction error rates compared to before imputation as can be seen (Fig 8). Their predictive performance exhibited that the log-rank split-rule is better than the log-rank-score split-rule. Moreover, we noticed that the Cox PH model showed good predictive performance compared to the two RSF under the two split-rules before and after imputation models. Thus it is safer to say that if all covariates satisfy the Cox PH assumption, the Cox PH model can be used [44].

Although the Cox PH model before and after imputation had better performance in terms of the prediction error rate, we can still not use it in the event of a violation of the proportionality of hazards assumption. Thus, in the presence of the non-proportional hazards genes/covariates, using RSF is an appealing option in the analysis of survival data, especially for high dimensional genomics data. Genomics data are usually presented in a matrix, with the columns indicating the samples and the rows showing a genomic feature such as genes [48].

Table 6 shows a comparison of the model performance using the integrated brier scores. We can notice that the prediction error estimates are lower for RSF, especially in the case of using the log-rank as a split rule. In addition, RSF models perform substantially better than Kaplan-Meier and Cox PH models.

Discussion

Cancer incidence and mortality are rapidly growing worldwide, exerting big physical, emotional, and financial problems on individual, families, communities, and health systems levels. Cancer is the first or second leading cause of death in 112 countries and is considered the third or fourth in 23 countries [3]. According to estimates from the World Health Organization (WHO), cancer is the leading cause of death around the world and accounting for nearly 10 million deaths in 2020. Moreover, WHO reported that CRC is the third common new cases, and it is also the second leading cause of death worldwide since 2020 [49]. The study aimed to determine the association between the genes and clinical covariates with CRC survival in the presence of missing values data. We also compared the predictive performance of the Cox PH and RSF models. The study provides essential information for CRC early detection and diagnosis.

The traditional regression-based methods to analyse survival data usually suffer from many problems such as restrictive assumptions including the proportionality, multicollinearity, curse of dimensionality, and lack of ability to rank the predictive performance. However, RSF models are frequently becoming a successful alternative for the analysis of the time to event data. In particular, the RSF is viewed as an appropriate analysing model for survival data, especially when the proportional hazards assumption is violated [39, 50]. When it comes to CRC survival analysis the gene expression and clinical information are utilized as covariates. The gene expression data contains many genes and most of these genes do not discriminate between normal cells and tumors. Therefore, we select the genes in which the change or difference in read counts between two conditions of experiment is statistically significant and such genes are known as the differentially expressed genes. In this study, the differentially expressed genes were obtained using three mutations based on the complete cases. The preliminary

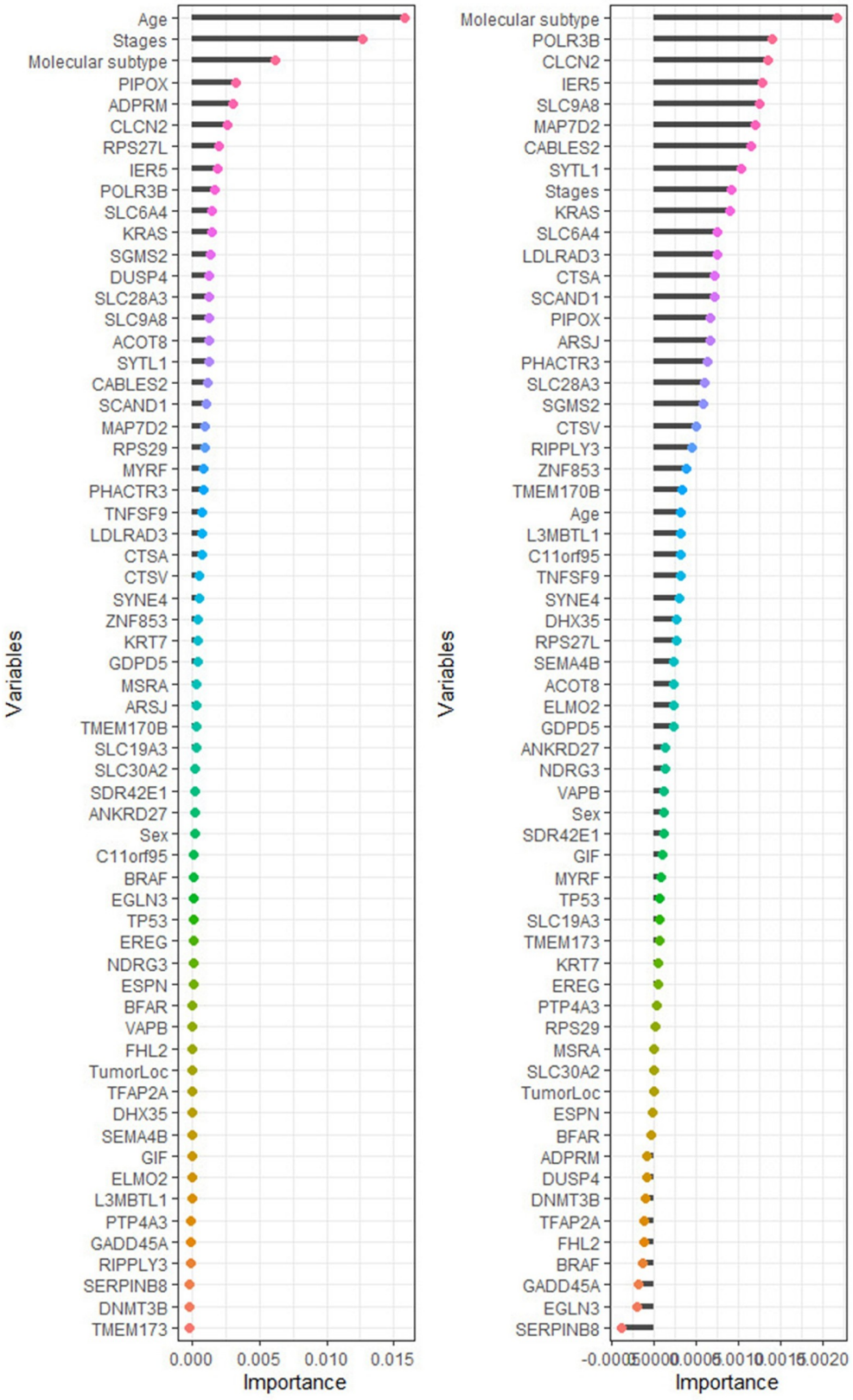

Age
Stages
Molecular subtype
PIPOX
ADPRM
CLCN2
RPS27L
IER5
POLR3B
SLC6A4
KRAS
SGMS2
DUSP4
SLC28A3
SLC9A8
ACOT8
SYTL1
CABLES2
SCAND1
MAP7D2
RPS29
MYRF
PHACTR3
TNFSF9
LDLRAD3
CTSA
CTSV
SYNE4
ZNF853
KRT7
GDPD5
MSRA
ARSJ
TMEM170B
SLC19A3
SLC30A2
SDR42E1
ANKRD27
Sex
C11orf95
BRAF
EGLN3
TP53
EREG
NDRG3
ESPN
BFAR
VAPB
FHL2
TumorLoc
TFAP2A
DHX35
SEMA4B
GIF
ELMO2
L3MBTL1
PTP4A3
GADD45A
RIPPLY3
SERPINB8
DNMT3B
TMEM173
Variables
0.000
0.005
0.010
0.015
Importance
Molecular subtype
POLR3B
CLCN2
IER5
SLC9A8
MAP7D2
CABLES2
SYTL1
Stages
KRAS
SLC6A4
LDLRAD3
CTSA
SCAND1
PIPOX
ARSJ
PHACTR3
SLC28A3
SGMS2
CTSV
RIPPLY3
ZNF853
TMEM170B
Age
L3MBTL1
C11orf95
TNFSF9
SYNE4
DHX35
RPS27L
SEMA4B
ACOT8
ELMO2
GDPD5
ANKRD27
NDRG3
VAPB
Sex
SDR42E1
GIF
MYRF
TP53
SLC19A3
TMEM173
KRT7
EREG
PTP4A3
RPS29
MSRA
SLC30A2
TumorLoc
ESPN
BFAR
ADPRM
DUSP4
DNMT3B
TFAP2A
FHL2
BRAF
GADD45A
EGLN3
SERPINB8
Variables
-0.0005
0.0000
0.0005
0.0010
0.0015
0.0020
Importance

Fig 6. The rank of most predictive genes and clinical variables for colorectal cancer patients' survival after the imputation is based on how they influence the survival outcome. The variables importance is built using log-rank and log-rank-score split-rules in the left and right panel, respectively.

https://doi.org/10.1371/journal.pone.0261625.g006

analysis showed that 54 potentially differentially expressed genes could be correlated with CRC survival and important for understanding the initiation and progression of CRC. The differentially expressed genes together with the clinical data were used to compare the predictive performance of the Cox PH model and RSF model before and after imputation on the CRC gene expression data.

We used stepwise regression for developing the Cox PH model at a 5% threshold level to get a simple model capturing the association between the top genes and CRC patient survival. Only five genes did not violate the Cox PH assumption in the final Cox PH model. The results show that the error rates of the RSF before imputation are much higher than the error rates for RSF built after imputation. Thus, the imputation can improve the performance of RSF. Although the Cox PH model had a better performance than RSF, the results from the current study demonstrate that the random survival forests models are more flexible than the models based on the Cox PH assumption as a prerequisite for variable inclusion in the model.

After imputation, the Cox PH model indicated *SLC9A8* and *ANKRD27* genes were no longer significant predictors of CRC survival. This because it is expected that the number of observations to increase, hence, statistical power to detect an effect. The variables that were not statistically significant before imputation may now be seen as statistically significant and vice versa. Therefore, this might affect the statistical power of some variables after imputation. Overall, the most prominent finding to emerge from the analysis based on Cox PH is that for one year increase in age, the hazards of death increase by 1.03, also the males are the most exposed to the hazards of death compared to females. Thus, this study supports evidence from previous observations [51–55].

The results of the RSF using both split-rules before and after imputation identified other genes/ covariates such as molecular subtype, *SLC6A4*, *KRAS*, *SGMS2*, *DUSP4*, and *SLC28A3*. These genes/ covariates show up as important in explaining CRC survival rates. However, these genes/ covariates did not appear very strongly associated with CRC survival in the Cox PH model. Thus, one interesting finding to note is that RSF models give additional information about variable importance.

Furthermore, the results from the two RSF models before and after imputation show that age, stages, molecular subtype, *SLC9A8*, *IER5*, *ARSJ*, *ANKRD27*, and *PIPOX* greatly affected

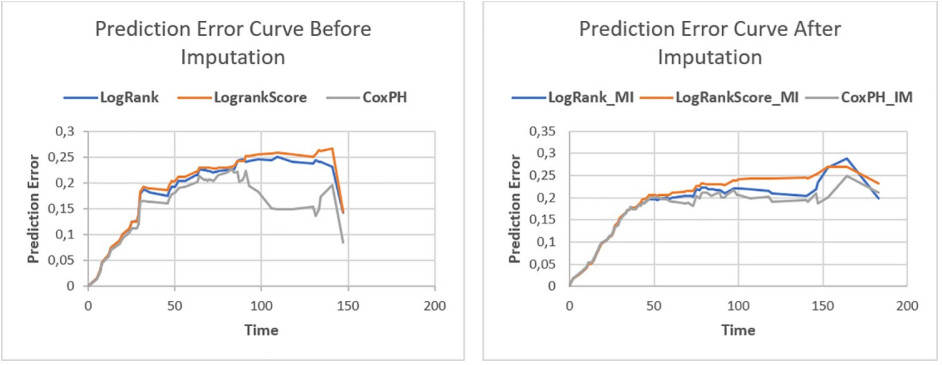


Fig 7. RSF with (log-rank and log-rank score) and Cox PH prediction error curve using 20% test set. The complete case and imputed dataset plots are in the left and right panel, respectively.

https://doi.org/10.1371/journal.pone.0261625.g007

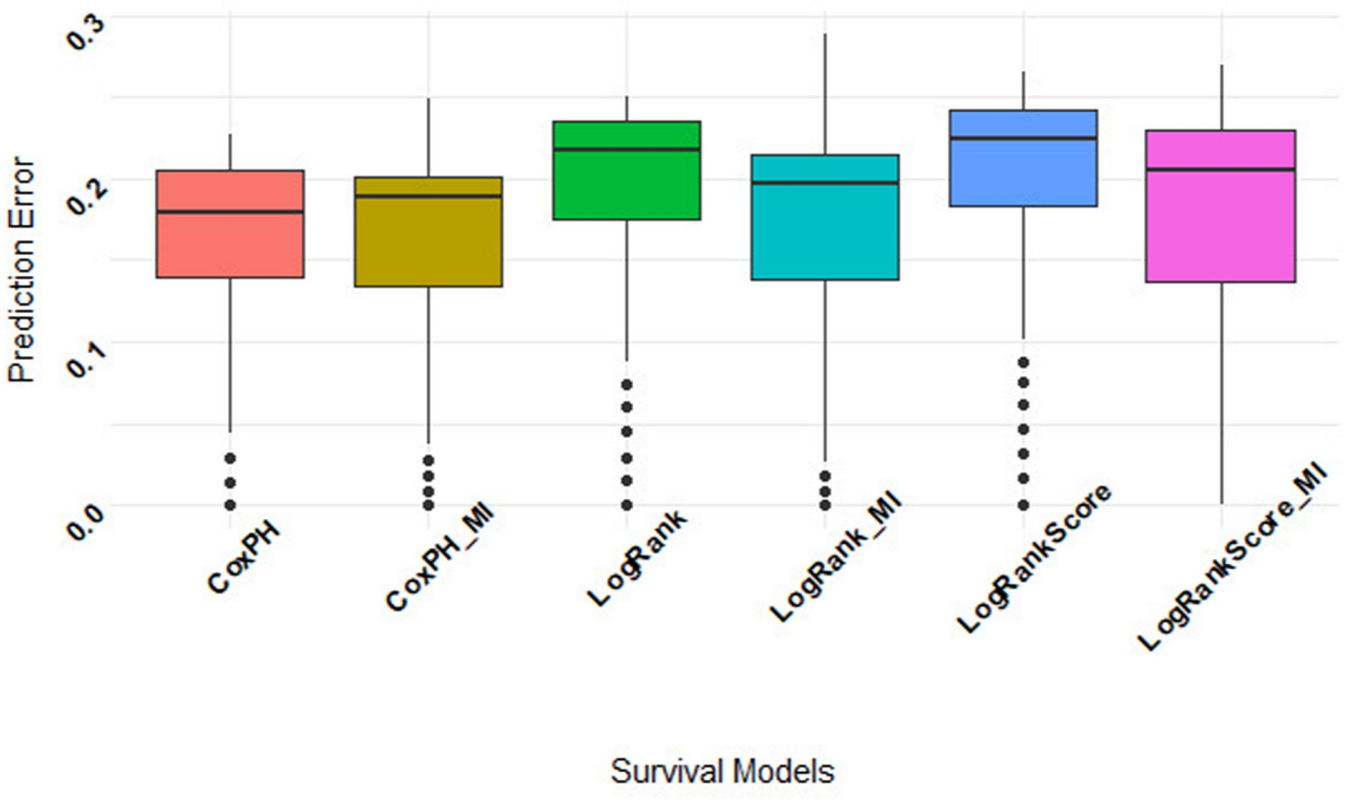


Fig 8. RSF with (log-rank and log-rank score) and Cox PH boxplot prediction error using 20% testing set together with the complete case dataset and the imputed data.

https://doi.org/10.1371/journal.pone.0261625.g008

the CRC mortality rates. These are ranked in the top 20 variables important in the two RSF models and agree with the Cox PH model results. Contrary to expectations, the RSF model did not pick sex as an important variable, while it is significant in the Cox PH model.

The Cox PH model had a better predictive performance in the presence of only those covariates that satisfy the Cox PH assumption compared to the RSF models. This result provides further support for the hypothesis that the Cox PH model works best under this assumption. In contrast, the out-of-bag error rate for the RSF with (log-rank and log-rank-score) before imputation is higher than that after imputation. This result implies that the imputation of missing values is a critical step and enormously improves the model's performance.

The most striking result to emerge from the analysis of the RSF is that log-rank has a better performance compared to the log-rank-score split-rule [44]. However, with more survival trees the log-rank-score seems to be stabilize compared to a smaller number of survival trees.

Table 6. Comparison of the models using the integrated brier scores.

Methods	Before Imputation	After Imputation
Kaplan Meier	0.199	0.201
RSF (Log-rank)	0.192	0.198
RSF (Log-rank score)	0.198	0.202
Cox PH	0.228	0.212

https://doi.org/10.1371/journal.pone.0261625.t006

We presented the development and validation of a robust five-gene signature (*SLC9A8*, *IER5*, *ARSJ*, *ANKRD27*, and *PIPOX*), which predicted overall survival (OS) for CRC patients. This gene signature was captured using Cox PH and RSF models based on two different scenarios. However, our study results successfully confirmed genes (markers) associated with CRC directly and identified new markers to enrich the field's literature further. Furthermore, the results support previous studies such as Mohammed *et al.* [56], where age, sex, and stages were also shown to be related to CRC survival.

Conclusion

Colorectal cancer (CRC) is a major cause of morbidity and mortality worldwide annually, making CRC the fourth common cause of death from cancer. However, the incidence of CRC has been steadily growing around the world, especially in developing countries. Therefore, the recent advances in technologies such as microarrays allowed for early detection screening using the individual's gene expression profiles.

The present study was designed to identify the genes prognosis of CRC. We developed a robust gene marker associated with the CRC overall survival based on gene expression data generated from microarray, using Cox PH and RSF models before and after missing data imputation. The most prominent finding to emerge from this study is that the Cox PH model identified five genes (*SLC9A8*, *IER5*, *ARSJ*, *ANKRD27*, and *PIPOX*) related to CRC overall survival in addition to age, sex (after imputation), and clinical stages. The RSF model further confirmed these results and had five additional gene markers predicting CRC survival. In addition, imputation improved the model's performance, and the current findings support the relevance of the missing data imputation. In summary, we recommend using a random survival forests model for survival data, especially in the high dimensional data where many genes might violate the Cox PH assumption.

Supporting information

S1 Appendix. Summary statistics of the 54 genes selected for survival analysis.
(PDF)

Acknowledgments

The authors wish to acknowledge and thank Dr. Justine B Nasejje for her helping in understanding and implementing the random survival forest methods.

Author Contributions

Conceptualization: Mohanad Mohammed, Henry Mwambi, Bernard Omolo.

Data curation: Mohanad Mohammed, Innocent B. Mboya.

Formal analysis: Mohanad Mohammed, Innocent B. Mboya.

Methodology: Mohanad Mohammed, Innocent B. Mboya, Henry Mwambi, Murtada K. Elbashir, Bernard Omolo.

Software: Mohanad Mohammed.

Supervision: Henry Mwambi, Murtada K. Elbashir, Bernard Omolo.

Writing – original draft: Mohanad Mohammed.

Writing – review & editing: Mohanad Mohammed, Innocent B. Mboya, Henry Mwambi, Murtada K. Elbashir, Bernard Omolo.

References

1. Favoriti P, Carbone G, Greco M, Pirozzi F, Pirozzi RE, Corcione F. Worldwide burden of colorectal cancer: a review. Updates Surg. 2016; 68(1):7–11. Epub 2016/04/14. https://doi.org/10.1007/s13304-016-0359-y PMID: 27067591.
2. Society AC. Colorectal Cancer Facts & Figures 2020–2022. Published online. 2020: 48.
3. Sung H, Ferlay J, Siegel RL, Laversanne M, Soerjomataram I, Jemal A, et al. Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: a cancer journal for clinicians. 2021; 71(3):209–49. https://doi.org/10.3322/caac.21660 PMID: 33538338
4. Dai GP, Wang LP, Wen YQ, Ren XQ, Zuo SG. Identification of key genes for predicting colorectal cancer prognosis by integrated bioinformatics analysis. Oncology letters. 2020; 19(1):388–98. https://doi.org/10.3892/ol.2019.11068 PMID: 31897151
5. Bray F, Ferlay J, Soerjomataram I, Siegel RL, Torre LA, Jemal A. Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: a cancer journal for clinicians. 2018; 68(6):394–424. https://doi.org/10.3322/caac.21492 PMID: 30207593
6. Stintzing S. Management of colorectal cancer. F1000Prime reports. 2014; 6:108. https://doi.org/10.12703/P6-108 PMID: 25580262
7. Bian Q, Chen J, Qiu W, Peng C, Song M, Sun X, et al. Four targeted genes for predicting the prognosis of colorectal cancer: A bioinformatics analysis case. Oncology letters. 2019; 18(5):5043–54. https://doi.org/10.3892/ol.2019.10866 PMID: 31612015
8. Society AC. Colorectal Cancer Early Detection, Diagnosis, and Staging. Published online. 2020: 40.
9. Molinari C, Marisi G, Passardi A, Matteucci L, De Maio G, Ulivi P. Heterogeneity in Colorectal Cancer: A Challenge for Personalized Medicine? International journal of molecular sciences. 2018; 19(12):3733. https://doi.org/10.3390/ijms19123733 PMID: 30477151
10. Bramsen JB, Rasmussen MH, Ongen H, Mattesen TB, Ornтoft MW, Arnadottir SS, et al. Molecular-Subtype-Specific Biomarkers Improve Prediction of Prognosis in Colorectal Cancer. Cell reports. 2017; 19(6):1268–80. https://doi.org/10.1016/j.celrep.2017.04.045 PMID: 28494874
11. Ogino S, Nowak JA, Hamada T, Phipps AI, Peters U, Milner DA Jr., et al. Integrative analysis of exogenous, endogenous, tumour and immune factors for precision medicine. Gut. 2018; 67(6):1168–80. https://doi.org/10.1136/gutjnl-2017-315537 PMID: 29437869
12. Abdul Aziz NA, Mokhtar NM, Harun R, Mollah MM, Mohamed Rose I, Sagap I, et al. A 19-Gene expression signature as a predictor of survival in colorectal cancer. BMC medical genomics. 2016; 9(1):1–13. https://doi.org/10.1186/s12920-016-0218-1 PMID: 27609023
13. Pan F, Chen T, Sun X, Li K, Jiang X, Försti A, et al. Prognosis prediction of colorectal cancer using gene expression profiles. Frontiers in oncology. 2019; 9:252. https://doi.org/10.3389/fonc.2019.00252 PMID: 31024853
14. Martinez-Romero J, Bueno-Fortes S, Martin-Merino M, Ramirez de Molina A, De Las Rivas J. Survival marker genes of colorectal cancer derived from consistent transcriptomic profiling. BMC Genomics. 2018; 19(8):45–60. https://doi.org/10.1186/s12864-018-5193-9 PMID: 30537927
15. Yan Z, Li J, Xiong Y, Xu W, Zheng G. Identification of candidate colon cancer biomarkers by applying a random forest approach on microarray data. Oncology reports. 2012; 28(3):1036–42. https://doi.org/10.3892/or.2012.1891 PMID: 22752057
16. Marisa L, de Reynies A, Duval A, Selves J, Gaub MP, Vescovo L, et al. Gene Expression Classification of Colon Cancer into Molecular Subtypes: Characterization, Validation, and Prognostic Value. Plos Medicine. 2013; 10(5):e1001453. https://doi.org/10.1371/journal.pmed.1001453 PMID: 23700391
17. Simon R, Lam A, Li M-C, Ngan M, Menenzes S, Zhao Y. Analysis of gene expression data using BRB-array tools. Cancer informatics. 2007; 3:11–7. PMID: 19455231
18. Chaba L, Odhiambo J, Omolo B. Evaluation of methods for gene selection in melanoma cell lines. International Journal of Statistics in Medical Research. 2017; 6(1):1–9. http://dx.doi.org/10.6000/1929-6029.
19. Bolstad BM, Irizarry RA, Astrand M, Speed TP. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. Bioinformatics. 2003; 19(2):185–93. https://doi.org/10.1093/bioinformatics/19.2.185 PMID: 12538238

20. Mohammed M, Mwambi H, Omolo B, Elbashir MK. Using stacking ensemble for microarray-based cancer classification. 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE): IEEE; 2018. p. 1–8.

21. Pedersen AB, Mikkelsen EM, Cronin-Fenton D, Kristensen NR, Pham TM, Pedersen L, et al. Missing data and multiple imputation in clinical epidemiological research. Clinical epidemiology. 2017; 9:157–66. https://doi.org/10.2147/CLEP.S129785 PMID: 28352203

22. Hayati Rezvan P, Lee KJ, Simpson JA. The rise of multiple imputation: a review of the reporting and implementation of the method in medical research. BMC medical research methodology. 2015; 15 (1):1–14. https://doi.org/10.1186/s12874-015-0022-1 PMID: 25880850

23. Mboya IB, Mahande MJ, Obure J, Mwambi HG. Predictors of perinatal death in the presence of missing data: A birth registry-based study in northern Tanzania. PLoS One. 2020; 15(4):e0231636. https://doi.org/10.1371/journal.pone.0231636 PMID: 32298332

24. Mboya IB, Mahande MJ, Obure J, Mwambi HG. Predictors of singleton preterm birth using multinomial regression models accounting for missing data: A birth registry-based cohort study in northern Tanzania. Plos one. 2021; 16(4):e0249411. https://doi.org/10.1371/journal.pone.0249411 PMID: 33793638

25. Jakobsen JC, Gluud C, Wetterslev J, Winkel P. When and how should multiple imputation be used for handling missing data in randomised clinical trials—a practical guide with flowcharts. BMC medical research methodology. 2017; 17(1):1–10. https://doi.org/10.1186/s12874-016-0277-1 PMID: 28056835

26. Azur MJ, Stuart EA, Frangakis C, Leaf PJ. Multiple imputation by chained equations: what is it and how does it work? International journal of methods in psychiatric research. 2011; 20(1):40–9. https://doi.org/10.1002/mpr.329 PMID: 21499542

27. Sterne JA, White IR, Carlin JB, Spratt M, Royston P, Kenward MG, et al. Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. BMJ. 2009; 338:b2393. https://doi.org/10.1136/bmj.b2393 PMID: 19564179

28. Bradburn MJ, Clark TG, Love SB, Altman DG. Survival analysis part II: multivariate data analysis—an introduction to concepts and methods. British journal of cancer. 2003; 89(3):431–6. https://doi.org/10.1038/sj.bjc.6601119 PMID: 12888808

29. Ajagbe OB, Kabir Z, O'Connor T. Survival analysis of adult tuberculosis disease. PLoS One. 2014; 9 (11):e112838. https://doi.org/10.1371/journal.pone.0112838 PMID: 25409024

30. Kleinbaum DG, Klein M. Survival analysis: Springer; 2010.

31. Myte R. Covariate selection for colorectal cancer survival data: A Comparison case study between random survival forests and the cox proportional-hazards model: Umeå University; 2013.

32. Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS. Random Survival Forests. The annals of applied statistics. 2008; 2(3):841–60. https://doi.org/10.1214/08-Aoas169

33. Wang H, Li G. A Selective Review on Random Survival Forests for High Dimensional Data. Quant Biosci. 2017; 36(2):85–96. https://doi.org/10.22283/qbs.2017.36.2.85 PMID: 30740388

34. Breiman L. Random forests. Machine Learning. 2001; 45(1):5–32. https://doi.org/10.1023/A:1010933404324

35. Jiang S. Prediction Based on Random Survival Forest. American Journal of Biomedical Science & Research. 2019; 6(2):109–11. https://doi.org/10.34297/ajbsr.2019.06.001005

36. Ishwaran H, Kogalur UB, Kogalur MUB. Package 'randomForestSRC'. 2021; 6:1–125.

37. Nelson W. Theory and applications of hazard plotting for censored failure data. Technometrics. 1972; 14(4):945–66. https://doi.org/10.2307/1267144

38. Ciampi A, Chang C-H, Hogg S, McKinney S. Recursive partition: A versatile method for exploratory-data analysis in biostatistics. Biostatistics: Springer; 1987. p. 23–50.

39. Nasejje JB, Mwambi H, Dheda K, Lesosky M. A comparison of the conditional inference survival forest model to random survival forests based on a simulation study as well as on two applications with time-to-event data. BMC medical research methodology. 2017; 17(1):1–17. https://doi.org/10.1186/s12874-016-0277-1 PMID: 28056835

40. Hothorn T, Lausen B. On the exact distribution of maximally selected rank statistics. Computational Statistics & Data Analysis. 2003; 43(2):121–37. https://doi.org/10.1016/S0167-9473(02)00225-6

41. Graf E, Schmoor C, Sauerbrei W, Schumacher M. Assessment and comparison of prognostic classification schemes for survival data. Statistics in medicine. 1999; 18(17–18):2529–45. https://doi.org/10.1002/(sici)1097-0258(19990915/30)18:17/18<2529::aid-sim274>3.0.co;2-5 PMID: 10474158

42. Ehrlinger J. ggRandomForests: Exploring random forest survival. arXiv preprint arXiv:161208974. 2016.

43. Taylor JM. Random Survival Forests. Journal of Thoracic Oncology. 2011; 6(12):1974–5. https://doi.org/10.1097/JTO.0b013e318233d835 PMID: 22088987

44. Nasejje JB, Mwambi H. Application of random survival forests in understanding the determinants of under-five child mortality in Uganda in the presence of covariates that satisfy the proportional and non-proportional hazards assumption. BMC research notes. 2017; 10(1):1–18. https://doi.org/10.1186/s13104-016-2345-3 PMID: 28057050

45. Gerds TA. Package 'pec'. R package version. 2020.

46. Team RC. R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing, 2021.

47. Chen G, Kim S, Taylor JM, Wang Z, Lee O, Ramnath N, et al. Development and validation of a quantitative real-time polymerase chain reaction classifier for lung cancer prognosis. Journal of Thoracic Oncology. 2011; 6(9):1481–7. https://doi.org/10.1097/JTO.0b013e31822918bd PMID: 21792073

48. Zang CZ, Wang T, Deng K, Li B, Hu SE, Qin Q, et al. High-dimensional genomic data bias correction and data integration using MANCIE. Nature Communications. 2016; 7(1):1–8. https://doi.org/10.1038/ncomms11305 PMID: 27072482

49. WHO. Cancer: WHO; 2021 [updated 03/03/2021; cited 2021 25/05/2021]. Available from: https://www.who.int/news-room/fact-sheets/detail/cancer.

50. Gerds TA, Kattan MW, Schumacher M, Yu C. Estimating a time-dependent concordance index for survival prediction models with covariate dependent censoring. Statistics in Medicine. 2013; 32(13):2173–84. https://doi.org/10.1002/sim.5681 PMID: 23172755

51. van Eeghen EE, Bakker SD, van Bochove A, Loffeld RJ. Impact of age and comorbidity on survival in colorectal cancer. Journal of gastrointestinal oncology. 2015; 6(6):605–12. https://doi.org/10.3978/j.issn.2078-6891.2015.070 PMID: 26697191

52. Jiang Z, Wang X, Tan X, Fan Z. Effect of Age on Survival Outcome in Operated and Non-Operated Patients with Colon Cancer: A Population-Based Study. PLoS One. 2016; 11(1):e0147383. https://doi.org/10.1371/journal.pone.0147383 PMID: 26789841

53. Chandrasinghe PC, Ediriweera DS, Nazar T, Kumarage S, Hewavisenthi J, Deen KI. Overall Survival of Elderly Patients Having Surgery for Colorectal Cancer Is Comparable to Younger Patients: Results from a South Asian Population. Gastroenterology Research and Practice. 2017; 2017:1–7. https://doi.org/10.1155/2017/9670512 PMID: 28811822

54. White A, Ironmonger L, Steele RJC, Ormiston-Smith N, Crawford C, Seims A. A review of sex-related differences in colorectal cancer incidence, screening uptake, routes to diagnosis, cancer stage and survival in the UK. BMC Cancer. 2018; 18(1):1–11. https://doi.org/10.1186/s12885-017-3892-2 PMID: 29291726

55. Abancens M, Bustos V, Harvey H, McBryan J, Harvey BJ. Sexual Dimorphism in Colon Cancer. Frontiers in Oncology. 2020; 10:1–27. https://doi.org/10.3389/fonc.2020.00001 PMID: 32076595

56. Mohammed M, Mwambi H, Omolo B. Colorectal Cancer Classification and Survival Analysis Based on an Integrated RNA and DNA Molecular Signature. Current Bioinformatics. 2021; 16(4):583–600. https://doi.org/10.2174/1574893615999200711170445