

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**MACHINE LEARNING BASED MODEL FOR CLINICAL
CHARACTERIZATION OF GERMLINE VARIANTS FROM RNA-
SEQ DATA**

ÖZGENUR AKDAŞ
A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE
DEPARTMENT OF BIOENGINEERING
BIOINFORMATICS AND SYSTEMS BIOLOGY PROGRAMME

GEBZE
2022

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**MACHINE LEARNING BASED MODEL
FOR CLINICAL CHARACTERIZATION OF
GERMLINE VARIANTS FROM RNA-SEQ
DATA**

ÖZGENUR AKDAŞ

**A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE
DEPARTMENT OF BIOENGINEERING
BIOINFORMATICS AND SYSTEMS BIOLOGY PROGRAMME**

**THESIS SUPERVISOR
PROF. DR. FATİH ERDOĞAN SEVİLGİN**

**GEBZE
2022**

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

RNA-SEQ DATASINDAN GERM HATTI
VARYANTLARIN KLİNİK
KARAKTERİZASYONU İÇİN MAKİNE
ÖĞRENMESİ TABANLI MODEL

ÖZGENUR AKDAŞ
YÜKSEK LİSANS TEZİ
BİYOMÜHENDİSLİK ANABİLİM DALI
BİYOİNFORMATİK VE SİSTEM BİYOLOJİSİ PROGRAMI

DANIŞMANI
PROF. DR. FATİH ERDOĞAN SEVİLGEN

GEBZE
2022



YÜKSEK LİSANS JÜRİ ONAY FORMU

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 07/07/2022 tarih ve 2022/34 sayılı kararıyla oluşturulan jüri tarafından 25/07/2022 tarihinde tez savunma sınavı yapılan Özgenur Akdaş'ın tez çalışması Biyomühendislik Anabilim Dalı Biyoinformatik ve Sistem Biyolojisi Programında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI)

: Prof. Dr. Fatih Erdoğan Sevilgen

ÜYE

: Doç. Dr. Tunahan Çakır

ÜYE

: Prof. Dr. Ahmet Okay Çağlayan

ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun

...../...../..... tarih ve/..... sayılı kararı.

İMZA/MÜHÜR

SUMMARY

Clinical class of variants obtained by variant analysis are estimated computationally according to ACMG-AMP criteria or retrieved from databases such as ClinVar. Variants in the ClinVar database have been characterized either experimentally or computationally. However, not all variants are classified as pathogenic or benign. Those non-classified variants are referred to as Variant of Uncertain Significance (VUS) in the literature. Variants are prioritized by filtering based on certain criteria such as population frequency, exonic function clinical class of the variants and then interpreted by experts. VUSs are often not reported by clinicians due to prioritization and interpretation problems. Therefore, VUSs are the one of the reasons for the low diagnostic rate, especially in rare hereditary diseases. In recent years, many studies have been published demonstrating that VUSs can be prioritized using expression and splicing information obtained from RNA-Seq data. Within the scope of this study, the significance of RNA-Seq data on VUS characterization was investigated and a novel random forest classifier based VUS characterization model is proposed. During training of the model, we exploit splicing and expression features, in addition to features such as pathogenicity prediction scores and population frequency which are customarily used by variant classifiers in published studies. The resulting model provides, 98% mean accuracy for the model using 5-fold cross-validation and %88 accuracy for predicting pathogenicity of VUSs (specificity) on a holdout test data. However, due to the insufficient number of variants affecting splicing and expression, the very limited effect of RNA-Seq data on variant characterization was detected.

Key Words: Variant of Uncertain Significance (VUS), RNA Sequencing, Mendelian Disease, Hereditary disease, Machine Learning.

ÖZET

Varyant analizi sonucu bulunan varyantların klinik sınıfı, ACMG-AMP kriterlerine göre hesaplamalı olarak tahmin edilmekte veya ClinVar gibi veri tabanlarından elde edilmektedir. ClinVar veri tabanındaki varyantlar, deneysel veya hesaplamalı olarak karakterize edilmiştir. Bununla birlikte, tüm varyantlar patojenik veya iyi huylu olarak sınıflandırılmamıştır. Bu sınıflandırılmamış varyantlar literatürde belirsiz öneme sahip varyant (Variant of Uncertain Significance, VUS) olarak adlandırılmaktadır. Varyantlar, varyantların popülasyon sıklığı, ekzonik fonksiyonu klinik sınıfı gibi belirli kriterlere göre filtreleme yapılarak önceliklendirilmekte ve uzmanlar tarafından yorumlanmaktadır. VUS'lar, önceliklendirme ve yorumlama sorunları nedeniyle klinisyenler tarafından sıklıkla raporlanmamaktadır. Bu nedenle VUS'lar, özellikle nadir görülen kalıtsal hastalıklarda düşük tanı oranlarının nedenlerinden biridir. Son yıllarda, RNA-Seq verilerinden elde edilen ekspresyon ve uç birleştirme (splicing) bilgileri kullanılarak VUS'ların önceliklendirilebileceğini gösteren birçok çalışma yayınlanmıştır. Bu çalışma kapsamında, RNA-Seq verilerinin VUS karakterizasyonu üzerindeki önemi araştırılmış ve yeni bir rastgele orman sınıflandırıcısı tabanlı VUS karakterizasyon modeli önerilmiştir. Modelin eğitimi sırasında, yayınlanmış çalışmalardaki varyant sınıflandırıcılar tarafından geleneksel olarak kullanılan patojenite tahmin puanları ve popülasyon sıklığı gibi özniteliklere ek olarak uç birleştirme ve ekspresyon özniteliklerinden faydalanılmıştır. Elde edilen model, 5 kat çapraz doğrulama kullanan model için %98 ortalama doğruluk ve bir test verisi üzerinde VUS'ların patojenitesini (özgüllük) tahmin etmek için %88 doğruluk sağlamaktadır. Ancak, uç birleştirme ve ekspresyonu etkileyen varyant sayısının yetersiz olması nedeniyle, RNA-Seq verilerinin varyant karakterizasyonu üzerindeki çok sınırlı etkisi tespit edildi.

Anahtar Kelimeler: Belirsiz Öneme Sahip Varyant, RNA dizileme, Mendeliyen hastalık, Kalıtsal hastalık, Makine Öğrenimi.

ACKNOWLEDGEMENTS

To begin with I would like to thank my thesis advisor, Fatih Erdoğan SEVİLGİN, for his guidance, support, and enlightening knowledge.

I would like to thank Tunahan ÇAKIR and Saliha DURMUŞ who are among the founders of the Bioinformatics and Systems Biology program, for creating an environment where we can study computational biology, as well as for their guidance, support and suggestions during the development of my thesis.

I am thankful to my colleagues at PHITech Bioinformatics; İhsan OVALI, Ünzile Güven GÜLHAN and Canan KOLAKOĞLU for their support and creating such a pleasant working environment. I would also like to thank Çiğdem Hazal TİMÜÇİN, my colleague, with whom I previously worked at PHITech Bioinformatics, for her support and help.

Finally, I would like to thank my family for their continuous encouragement and support.

TABLE of CONTENTS

	<u>Page</u>
SUMMARY	v
ÖZET	vi
ACKNOWLEDGMENTS	vii
TABLE of CONTENTS	viii
LIST of ABBREVIATIONS and ACRONYMS	x
LIST of FIGURES	xi
LIST of TABLES	xii
1. INTRODUCTION	1
2. TRANSCRIPTOME (RNA) SEQUENCING	5
2.1. Next Generation Sequencing	7
3. BIOINFORMATIC ANALYSES OF RNA-SEQ DATA	10
3.1. Variant Calling from RNA-Seq Data	15
3.1.1. Variant Annotation	17
3.2. Expression Analysis from RNA-Seq Data	21
3.3. Splicing Analysis from RNA-Seq Data	23
4. ML BASED STRATEGIES FOR CLINICAL CHARACTERIZATION OF GERMLINE VARIANTS	27
4.1. Machine Learning Based Approaches	27
4.2. Random Forest Model	29
4.3. Model Performance Metrics	32
5. MODEL AND DATASET CONSTRUCTION	34
6. FEATURE ENGINEERING	42
7. RESULT AND DISCUSSION	45
7.1. Feature Importance	45
7.2. Significance of Expression on Variant Characterization	53
7.3. Significance of Splicing on Variant Characterization	55
7.4. Model Comparison and Handling Imbalanced Data	56
8. CONCLUSION AND FUTURE WORK	62

REFERENCES

64

BIOGRAPHY

72



LIST OF ABBREVIATIONS AND ACRONYMS

<u>Abbreviations</u>	<u>Explanations</u>
<u>and Acronyms</u>	
Alt allele	: Alternate allele
BAM	: Binary Alignment Map
BWT	: Burrows-Wheeler Transform
CF	: Cystic Fibrosis
DEG	: Differential Expressed Gene
DNA	: Deoxyribonucleic Acid
FAP	: Familial Adenomatous Polyposis
FPKM	: Fragments Per Kilobase Million
GATK	: Genome Analysis Toolkit
GEO	: Gene Expression Omnibus
INDEL	: Insertion and Deletion
LS	: Lynch Syndrome
mRNA	: Messenger Ribonucleic acid
NGS	: Next Generation Sequencing
PCR	: Polymerase Chain Reaction
pre-mRNA	: Precursor Messenger Ribonucleic acid
Ref allele	: Reference allele
RNA-Seq	: Ribonucleic acid Sequencing
RPKM	: Reads Per Kilobase Million
rRNA	: Ribosomal Ribonucleic acid
SAM	: Sequence Alignment Map
SNV	: Single Nucleotide Variant
TPM	: Transcript Per Million
tRNA	: Transfer Ribonucleic acid
VCF	: Variant Call Format
VUS	: Variant of Uncertain Significance
WES	: Whole Exome Sequencing
WGS	: Whole Genome Sequencing

LIST OF FIGURES

<u>Figure No:</u>	<u>Page</u>
2.1: Central Dogma.	5
2.2: Four lines in the FASTQ file.	7
2.3: NGS general workflow.	8
3.1: FastQC output, per base sequence quality.	11
3.2: Alignment to the reference genome.	11
3.3: Seed searching step.	13
3.4: Clustering/stitching/scoring step.	13
3.5: Columns in a variant call record.	16
3.6: Common alternative splicing events.	24
3.7: Canonical splice site.	24
3.8: Part of the annotation file.	26
4.1: Model selection in machine learning.	29
4.2: An example of decision tree.	30
4.3: An example of a random forest.	31
4.4: Confusion matrix and performance metrics.	32
5.1: Summary of dataset construction steps.	41
7.1: Feature importance charts for cat_FC1.	46
7.2: Feature importance charts for cat_FC2.	46
7.3: Feature importance charts for cat_FC.	47
7.4: Feature importance of the model trained with balanced and imbalanced dataset using features in VUS_char.	49
7.5: Feature importance of the model trained with balanced and imbalanced dataset using features in VUS_char_expression.	50
7.6: Feature importance of the model trained with balanced and imbalanced dataset using features in VUS_char_splicing.	52
7.7: Confusion matrix of the model trained with class weight = none.	57

LIST OF TABLES

<u>Table No:</u>	<u>Page</u>
3.1: Mandatory fields in the SAM format and their descriptions.	14
3.2: Description of columns in a vcf file.	16
3.3: Databases for gene based annotation in ANNOVAR.	18
3.4: Databases for filter based annotation in ANNOVAR.	19
3.5: The part of raw count matrix.	22
5.1: Accession IDs and diseases.	34
5.2: SpliceAI delta scores and categorization of scores.	37
5.3: An example feature table.	38
5.3: Continuation.	39
5.4: Datasets and variant count.	40
6.1: FC and categorical expression values.	43
7.1: Feature group name and features.	47
7.1: Continuation.	48
7.2: Accuracy values of the models trained with various feature groups.	48
7.3: Performance metrics of the model trained with FC calculated using HPA tissue specific expression value.	53
7.4: Performance metrics of the model trained with FC calculated using expression value of controls in datasets.	54
7.5: Model performance metrics for splicing feature obtained by splicing analysis of variants in databases.	55
7.6: Comparison of predictions of one of the cross-validation models.	56
7.7: Performance metrics of the models trained by changing class weights.	59
7.8: Performance metrics of the models trained by changing class weights for hold-out data.	59
7.9: Performance metrics of the models trained with balanced data.	60
7.10: Performance metrics of the models trained with balanced data for hold-out data.	60

1. INTRODUCTION

Next Generation Sequencing (NGS) is a massively parallel and high-throughput sequencing technology. The NGS method has facilitated the sequencing of large amounts of genomic data in recent years. Whole Genome Sequencing (WGS), in which the entire DNA of an individual is sequenced, and Whole Exome Sequencing (WES), that targets only protein coding regions of the genome, are popular DNA based NGS methods. With the decrease in sequencing cost, these two DNA-based sequencing methods, mostly WES, have started to be used in the routine clinical practice. The purpose of these two DNA-based methods is to detect variants that are disease related. Variants are defined as differences between a representative sequence called reference genome and a sample. The raw data obtained via sequencing methods is processed with bioinformatic algorithms and a list of variants is generated. Information whether these variants are disease-related (pathogenic) or benign, in other words the information about clinical class of the variants, are obtained in the annotation step according to ACMG-AMP [Richards et al., 2015] criteria or using databases such as ClinVar [Landrum et al., 2014].

ACMG-AMP criteria which is published by the American College of Medical Genetics and Genomics (ACMG) and the Association for Molecular Pathology (AMP), consists of 28 criteria. In these criteria, many pieces of information such as the frequency of the variant in the population, the prediction scores published in the literature (SIFT, PolyPhen etc.), the effect on the protein are considered to evaluate variants' clinical class computationally. Proposed criteria are specific to variants detected in inherited disorders, especially mendelian diseases. The criteria are grouped as “very strong”, “strong”, “supporting” etc. for both pathogenic and benign cases. A variant is characterized by one of the five main classes according to the number of criteria met for each group. The main classes are “pathogenic”, “likely pathogenic”, “benign”, “likely benign” and “variant of uncertain significance”. For example, a variant is characterized as pathogenic if it meets more than two criteria which are in pathogenic strong (PS) group. If the variant cannot be classified in one of the main classes or classified conflictingly as both being pathogenic and benign, it is characterized as variant of uncertain significance (VUS).

Variants in the ClinVar database have been characterized based on experimental studies or computational evaluation using improved versions of ACMG-AMP criteria.

Characterized variants are submitted by different institutions conducting the study. Unlike ACMG-AMP, there are classes such as “risk factor” and “drug response” according to the relationship of the variant with the disease in the ClinVar database.

VUSs are variants whose impact on individual have not been determined and therefore cannot be classified as pathogenic or benign. The largest number of variants in the ClinVar database belongs to the variant of uncertain significance (VUS) class. There are a total of 1539888 variants in the ClinVar Miner database (as of July 2nd, 2022) and 674594 of them were classified as VUS [Henrie et al., 2018]

Prioritization is the process of obtaining variants that can be associated with the phenotype. Variants found in variant analysis are prioritized by filtering based on certain criteria such as population frequency and variant effect on the protein. The purpose of prioritization is to create a list of candidate variants to facilitate detection of the target variant causing the disease. Variant interpretation is evaluation of candidate variants by experts. For example, if a variant is rare in the population and pathogenicity scores such as SIFT and PolyPhen are estimated as deleterious, this variant is determined as a candidate variant that may be associated with the disease. By applying the same filters, the variants found as a result of the variant analysis are filtered and the candidate variant list is generated. The candidate variants are then interpreted together with the patient's phenotype as to whether they are the target variant associated with the disease. VUSs, on the other hand, cannot be prioritized and their association with the disease cannot be interpreted.

Although WES and WGS are used for disease diagnosis in routine clinical practice, many variants are found are VUSs [Kremer et al., 2017]. These variants mostly not reported by the clinicians due to prioritization and interpretation problems. Thus, many variants with the potential to cause disease are ignored. So, the diagnostic rate of WGS and WES technologies in rare diseases, remains at 50%. One of the most important reasons why the diagnosis rate in hereditary diseases is low, especially in rare diseases, is the failure to characterize VUSs.

RNA-Seq data is very rich in terms of information it contains, as it can be processed with many different analysis types. These types of analysis mainly include (i) measurement of gene expression levels, (ii) identification of genomic variants, (iii) identification of splicing events, and (iv) detection of gene fusions. Fusion gene detection from RNA-Seq data mainly performed for non-hereditary (somatic) diseases

[Kim et al., 2019]. Other analyses from RNA-Seq data are performed for both hereditary (germline) and non-hereditary (somatic) diseases.

Recent studies have revealed the power of RNA-Seq on variant interpretation and prioritization. In these studies, the possibility of VUSs to be pathogenic was evaluated according to the splicing and expression information obtained from RNA-Seq data. If a variant disrupts splicing or causes excessive increase or decrease in gene expression level, variant is more likely to be pathogenic. It was observed that the diagnosis rate by using only WGS and WES data is, increased by 10-35% with the contribution of RNA-Seq data. [Cummings et al., 2017; Kremer et al., 2017; Marco-Puche et al., 2019; Murdock et al., 2021; Yépez et al., 2022]

Several machine learning (ML) based tools are developed for the variant classification. LEAP variant classifier is specific to missense variants. The training set consists of variants in genes associated with hereditary cancer and cardiovascular diseases. Publicly available information such as functional impact predictors and splicing impact predictors are used to train Logistic Regression and Random Forest Classifier models [Lai et al., 2020]. Logistic regression model as a result of cross validation test achieved an area under the receiver operating characteristic curve (AUROC) of 97.8% (cancer) and 98.8% (cardiovascular), while the random forest model achieved 98.3% (cancer) and 98.6% (cardiovascular). Hold-out data was used to test the Leap models, as well. Gene-holdout predictions achieved 96.8% AUROC. In another ML based variant classifier called ClinPred, non-synonymous variants in the ClinVar database (January 2016 version) are used to train random forest and gradient boosted decision tree (XGBoost) models. Functional scores such as SIFT, PolyPhen, frequency and conservation scores are used as features during model development. The AUC score of the ClinPred model on the ClinVar test set is 0.98. It is important to note that, these two variant classifiers are DNA-Seq based models.

Here, we present the first variant classifier model based on machine learning using RNA-Seq data. To the best of our knowledge, there is no RNA-Seq-based variant classifier model, so far. In our study, we examined that the contribution of RNA-Seq VUS characterization ML models. Utilization of expression and splicing features data distinguishes the model from its counterparts. The model developed using public RNA-Seq data showed a performance of 98% accuracy and 88% accuracy for predicting pathogenicity of VUSs (specificity) on a holdout test data.

Although the performance of the proposed model is satisfactory, the contribution of RNA-Seq data to variant characterization has not been fully observed. This is basically due to the insufficient number of variants that are affected by the expression and splicing process.

Although several additional experiments were carried out to show the effect of expression, no improvement in model performance was observed. Moreover, there is no database containing the expression information of the genes in which the variants are located in. So, presenting the effect of expression value of the model requires further studies when more RNA-Seq data is available. On the other hand, to calculate splicing scores, the chromosome, position, ref allele and alt allele information of the variant are sufficient. To show the effect of splicing on the variant characterization within the scope of the thesis, an additional study was carried out using variants in databases. As a result, the effect of splicing on the variant characterization was partially demonstrated.

2. TRANSCRIPTOME (RNA) SEQUENCING

Ribonucleic acid (RNA) is a single-chain molecule found in the nucleus, cytoplasm, ribosome, and mitochondria in human cells. On the other hand, deoxyribonucleic acid (DNA) that is found in nucleus and stores genetic information of humans is made up of two strands, in which the base pairs complement each other as base Adenin (A) pairs with base Timin (T) and base Cytosine (C) pairs with base Guanin (G). RNA has a thymine base instead of adenine and a five-carbon ribose sugar instead of deoxyribose unlike DNA. RNAs are divided into two main classes, coding, and non-coding, according to their potential to be translated into proteins. Messenger RNAs (mRNAs) are the coding RNAs. Non-coding RNAs are functional but cannot translated into protein. They are either structural (carrier RNA, ribosomal RNA) or regulatory RNAs. Regulatory RNAs are further divided into two types as non-protein coding RNAs shorter than 200 nucleotides (microRNA) and longer than 200 nucleotides (lncRNA) [Dahariya et al., 2019].

The entire process of transforming the genetic information in DNA into the functional product (protein) via RNA is called central dogma by Crick in 1958 [CRICK, 1970]. As shown in Figure 2.1, this process consists of two main stages: transcription and translation. Transcription is the process of creating mRNA from DNA. Translation (reading) is the process of synthesizing protein from mRNA.

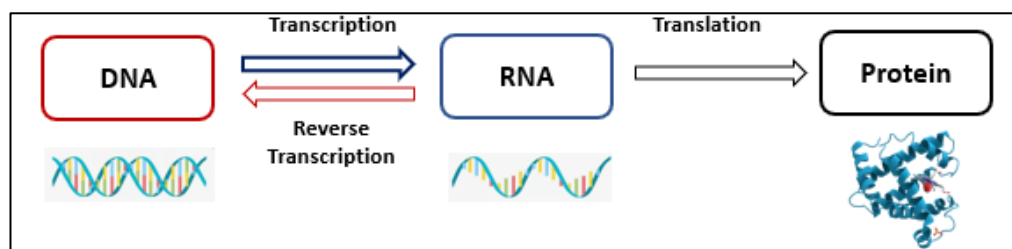


Figure 2.1: Central Dogma.

There are three main types of RNA involved in protein synthesis. These are messenger RNA (mRNA), transfer RNA (tRNA), and ribosomal RNA (rRNA).

Messenger RNA (mRNA) is a type of RNA in which the genetic information in DNA is copied. The order of the bases on the messenger RNA gives the amino acid sequence in the protein to be synthesized. The transcription process takes place in the cell nucleus of eukaryotic organisms such as human. The initially synthesized RNA in

eukaryotic organisms is called pre-mRNA (pre-mRNA), and it must go through certain stages to turn into mature mRNA which is used in translation process. In non-eukaryotic organisms, on the other hand, mRNA transcribed from DNA mostly participates directly in the protein synthesis process and does not require any additional processing.

Transfer RNA (tRNA) is the RNA with the smallest size among the three main types of RNA. It is called carrier RNA because it is responsible for the transport of amino acids during protein synthesis.

Ribosomal RNA (rRNA) is a type of RNA that makes up 80% of all RNA in a cell and is found in ribosomes. Ribosomal RNA oversees acting like an enzyme, linking the amino acids brought by the tRNA to the polypeptide chain that will form the protein.

The transcriptome is the whole RNA transcripts in a sample, including protein-coding and other non-protein-coding transcripts. Transcriptome sequencing or RNA sequencing (RNA-Seq) is technology that provides comprehensive analysis of the transcriptome in a sample. Standard transcriptome sequencing steps are isolation of all RNA strands in the cells, selection of RNA molecule to be sequenced, conversion from RNA to cDNA, adapter ligation, amplification, and sequencing. The steps from conversion to cDNA to amplification is called library preparation. RNA isolation is the first step in transcriptome sequencing. The quality of the isolated RNA should be checked before proceeding to the sequencing step. The often-preferred criterion for assessing RNA quality is the RNA integrity number (RIN) [Schroeder et al., 2006]. RIN produces values between 1 and 10. The closer this value is to 10, the higher the quality of the RNA (Gallego Romero et al., 2014). Isolates are fragmented before being converted to cDNA. These fragments are then converted to cDNA, which is more stable than RNA. Adapter sequences that specific to the kit and platform are ligated to the fragments. Adaptor sequences mainly used for binding of fragments to the flow cells. After the fragments are amplified by methods such as polymerase chain reaction (PCR), the targeted RNA is sequenced.

The output obtained from the sequencing step is a FASTQ formatted file which is used in bioinformatic analysis afterwards. Each biological sequence in the FASTQ file is referred to as read. Each read in this file consists of four lines. The first line is the sequence identifier, the second line is the raw biological sequence, the third line

starting with + space is used for any description and the last line the quality scores of each letter in the raw sequence, as shown in Figure 2.2.

1	@ERR000589.41 EAS139_45:5:1:2:111/1
2	CTTTCCTCCCTGCTTTCCTGGCCCCACCATTTCCAGGGAACATCTTGTCAT
3	+
4	3IIIIIIIIIIII>1IIIF9BG08E00I%IG+&?(4)%00646.C1#&(

Figure 2.2: Four line in the FASTQ file.

Quality scores on the last line are often expressed in ASCII characters. Quality scores are commonly known as the phred or Q score. The phred quality score is calculated according to the Formula in 2.1. P represent the error probability of a nucleotide to be sequenced incorrectly. If the phred quality score is 20, it means that the relevant base is 99% correct according to this formula. If the probability of error is 0.01, the ASCII character of “5” is used for the phred score of 20. The characters with lower ASCII codes have higher probability of error while the characters with higher ASCII codes have lower probability.

$$Q_{\text{Phred}} = -10 * \log_{10}P \quad (2.1)$$

2.1. Next Generation Sequencing

NGS (Next Generation Sequencing) is a sequencing technology known as second generation sequencing as well. First generation is the "chain termination method" (or Sanger method) for determining the nucleotide sequence of DNA or RNA, developed by Frederick Sanger and his colleagues in 1977 [Sanger et al., 1977].

In principle, the concepts behind Sanger and next-generation sequencing (NGS) technologies are similar. In both NGS and Sanger sequencing (also known as dideoxy or capillary electrophoresis sequencing), DNA polymerase adds own fluorescent nucleotides one at a time to a growing DNA sequence. Each included nucleotide is identified by its fluorescent label. The critical difference between Sanger sequencing and NGS is the sequencing size. While the Sanger method sequences only a single piece of DNA at a time, NGS is largely parallel. So, sequencing millions of pieces simultaneously at each run is possible [Slatko et al., 2018].

NGS methods are divided into deoxyribonucleic acid (DNA) and ribonucleic acid (RNA) sequencing according to the genetic material of the organism to be sequenced. DNA sequencing methods are divided into whole genome sequencing (WGS), whole exome sequencing (WES) and panel sequencing according to the targeted region. WGS is the method by which all the nucleotides that make up the genome are sequenced, which expresses the entire genetic information of an organism. WES is a method in which protein-coding regions corresponding to 2% of the genome are sequenced. Panel sequencing is the method that is mostly used to search for a specific mutation associated with the disease and to target certain genes. This method is cost effective compared to other DNA sequencing methods. If it is aimed to discover a new gene associated with a disease, other DNA sequencing methods with higher cost and lower sequencing depth should be preferred instead of panel sequencing.

In principle the general workflow of all NGS methods is the same and it is shown in the Figure 2.3.

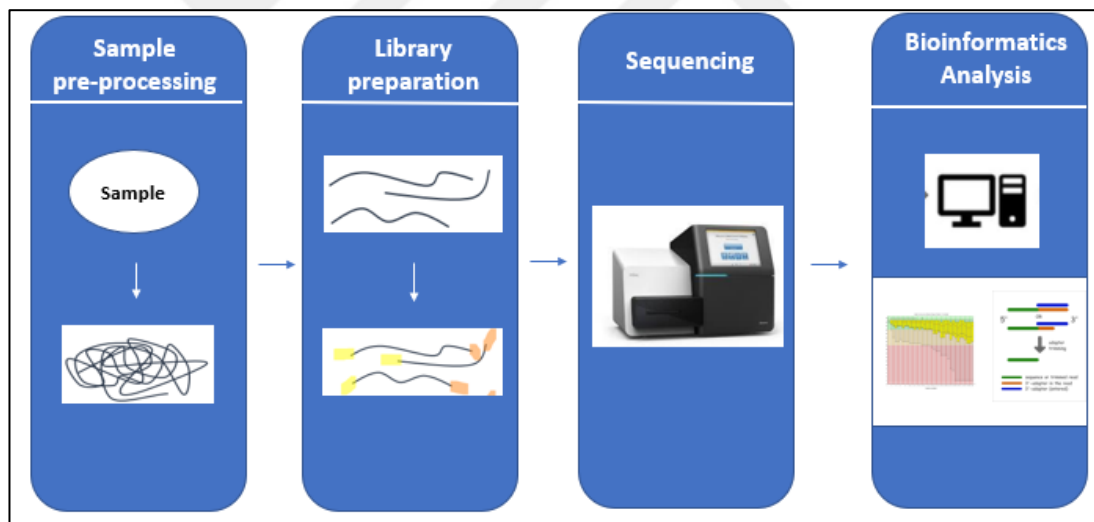


Figure 2.3: NGS general workflow.

NGS methods begin with sample isolation. Longer reads are obtained with NGS than with sanger sequencing, but NGS devices are also not capable of sequencing the entire sequence. Therefore, isolated DNA or RNA molecules are fragmented. The library preparation step consists of amplification and adapter ligation. Adaptors are short sequences of 25-30 nucleotides and are designed to complement sequences embedded on the flow cell in sequencing devices. In the adapter ligation step, the fragments can be attached to the flow cell by means of adapters that ligate the

fragments. The purpose of the amplification process is to increase the number of fragments, this process is performed by PCR. The library preparation step is followed by sequencing. The raw data used in the analysis and obtained by the sequencing is FASTQ file. The final step in the NGS workflow is bioinformatic analysis where the sequence is interpreted. For example, if WES is performed, the aim of the bioinformatic analysis would be discovery of disease-related variants.



3. BIOINFORMATIC ANALYSES OF RNA-SEQ DATA

Bioinformatic analysis is the process of extracting useful information from biological data. The analysis workflow or pipeline refers to all the steps performed in a certain order to process biological data.

First few basic steps of the bioinformatic analysis pipeline are common for both DNA and RNA-based data. These are data quality control, trimming, and alignment to the reference genome. However, RNA-based data provides more information over DNA data via downstream analysis. While variant detection is the main bioinformatic analysis type with genomic data, various analyses such as measurement of expression levels, detection of variants, and splicing are performed using RNA-Seq data.

The bioinformatics pipeline of NGS data begins with quality control of reads. The quality control (QC) process typically involves assessing the actual quality of raw reads using metrics generated by the sequencing platform (for example, quality scores) or calculated directly from raw reads. The purpose of the quality control step is to detect many sequencing artifacts such as base calling errors and primer/adaptor contamination [Patel and Jain, 2012]. The quality of data is crucially important for various downstream analyses, such as variant detection or gene expression studies. For example, if adapter sequences are not removed before variant analysis, they will be mapped to regions that are like these sequences in the reference genome and differences will be reported as fictional variants.

There are many tools for assessing the quality of NGS reads such as NGS QC Toolkit [Patel and Jain, 2012], NGSQC [Dai et al., 2010]. The most popular one is FastQC that published by Babraham Institute [Trivedi et al., 2014]. FastQC tool reports basic statistics and metrics such as base quality, overrepresented sequences, adapters, GC content. Duplication, k-mer or GC content levels in the FastQC output depend on experiment and organism [Conesa et al., 2016].

Poor quality reads need to be removed besides adapters and barcodes. Common tools for this trimming operation are Trimmomatic [Bolger et al., 2014] and Cutadapt [Martin, 2011]. Figure 3.1 demonstrates an example per base quality report provided by FastQC. The y-axis on the graph represents quality score. Higher scores refer to the better base call. FastQC reports a failure if the median for any base is less than 20. These bases must be removed from the reads.

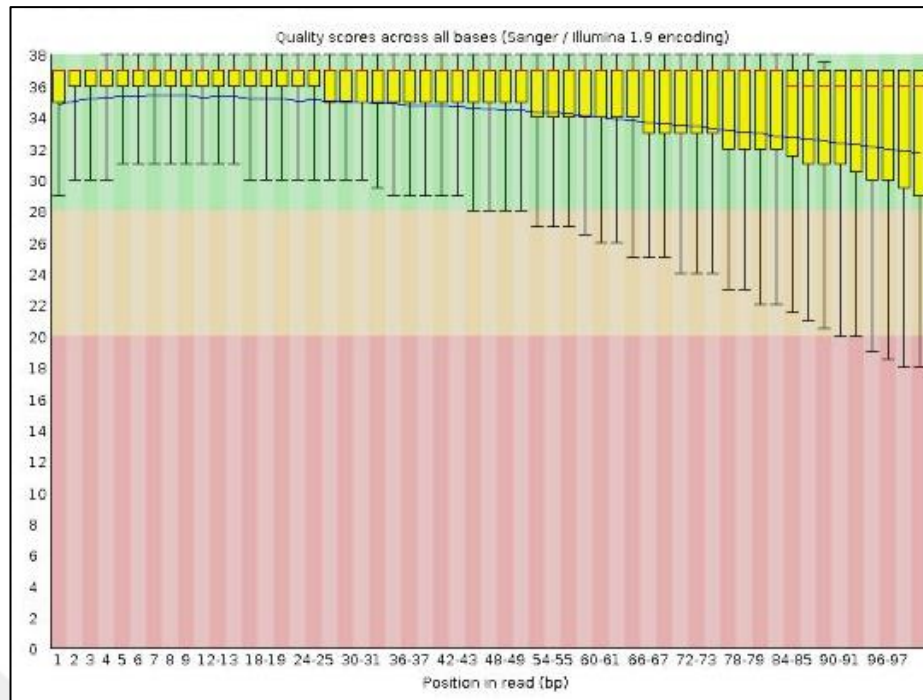


Figure 3.1: FastQC output, per base sequence quality.

Clean reads that are obtained after trimming are aligned to the reference genome. The reference genome is the representative, ideal DNA sequence of all individuals of the species. The last version GRCh38 was used in the GENCODE (GENCODE v36) [Frankish et al., 2019] within the scope of the thesis. As shown in Figure 3.2, alignment or mapping is the process of obtaining a complete genome of an organism by putting together the matching parts of the reference genome, like a jigsaw puzzle, of the clean reads. The actual locations of the reads from the sequencing machine are tried to be found in the reference genome.

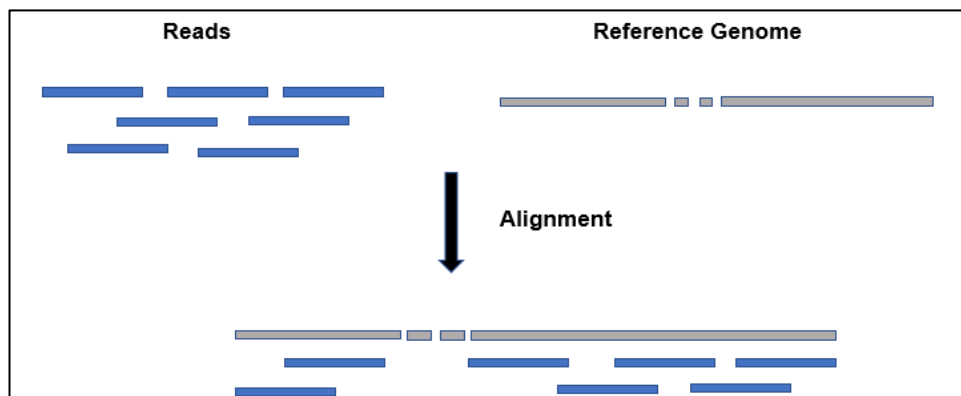


Figure 3.2: Alignment to the reference genome.

Many aligner tools are available for use in NGS data analysis. Some of the criteria that should be considered when selecting the appropriate aligner that affect the performance of the aligner are as follows, i) length of reads obtained after sequencing, ii) sequencing quality, iii) errors due to sequencing, iv) memory requirement, v) sequencing depth [Keel and Snelling, 2018]. In the light of these criteria, the appropriate algorithm for analysis and the tool that adopts this algorithm are selected. The alignment algorithms can be divided into two main groups, called as global alignment and local alignment. The purpose of the global alignment algorithm is to align over the entire sequence, and the purpose of the local alignment is to find the regions with most similarity and align to those regions. The most well-known examples of these algorithms are Needleman-Wunsch [Needleman and Wunsch, 1970] and Smith-Waterman [Smith and Waterman, 1981] respectively. Many aligners including Bowtie [Langmead et al., 2009] and BWA [Li and Durbin, 2009] have been developed based on the theory of Burrows–Wheeler Transform (BWT). BWA and Bowtie tools are designed for alignment of DNA sequences. Splice aware aligner is needed for alignment of transcripts since tools like BWA do not handle intron-sized gaps [Baruzzo et al., 2017; Musich et al., 2021]. Two of the well-known and mostly used transcriptomic aligners are HISAT [Kim et al., 2015] and STAR [Dobin et al., 2013]. STAR aligner was used for RNA-Seq analysis performed within the scope of the thesis.

The reference genome must be indexed before the alignment with the STAR tool, as with other alignment tools. STAR alignment algorithm consists of seed searching and clustering/stitching/scoring steps. In the seed searching step, STAR searches for the longest sequence that exactly matches one or more locations. The longest matching sequences are called Maximal Mappable Prefixes. The part of the read that aligns with the reference genome is called the seed. Figure 3.3 demonstrates seed searching step.

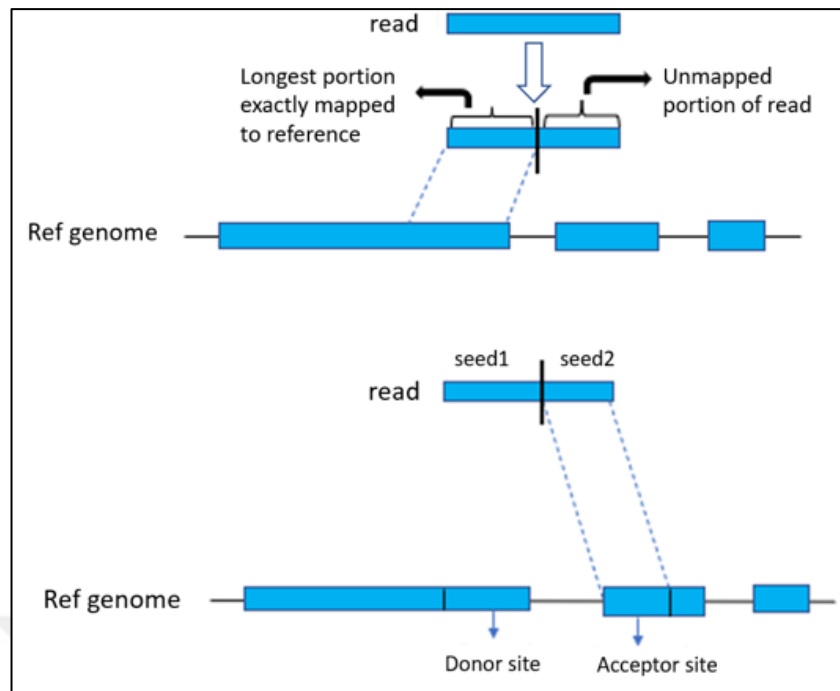


Figure 3.3: Seed searching step.

The seeds are clustered and scored based on their proximity to the seed in the original read, and scored based on mismatches, indels, gaps after seed searching step. Seeds showing the best alignment for reading are stitched as shown in Figure 3.4.

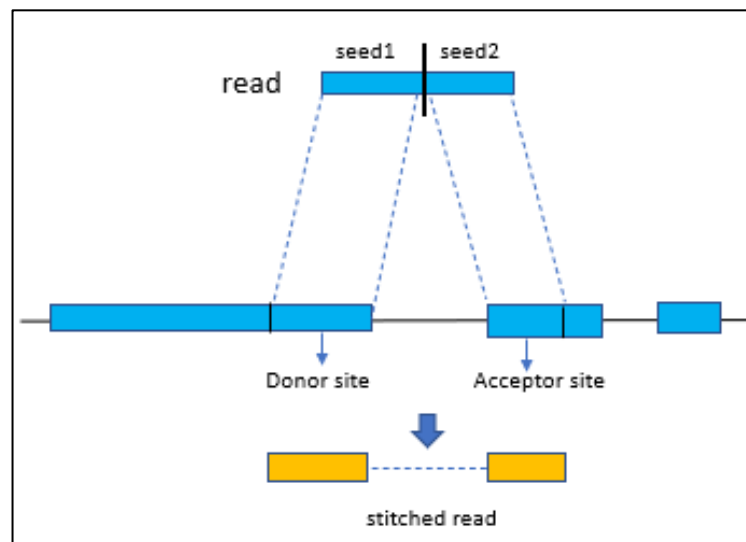


Figure 3.4: Clustering/stitching/scoring step.

At the end of the alignment step, the sequences mapped to the reference genome are stored in the Sequence Alignment Map (SAM), a text-based file. The file in SAM

format consists of header and alignment parts. The parts that start with “@” in the SAM file represent the headers. Each line in the alignment section represents the alignment of a segment and consists of eleven or more fields. The first eleven fields are always present in a SAM file. The first eleven columns in Table 3.1 are mandatory for the alignment section. Description in the Table 3.1 acquired from Samtools Github website [Sequence Alignment/Map Format Specification, 2021].

Table 3.1: Mandatory fields in the SAM format and their descriptions.

Field	Description
QNAME	Query name (read ID).
FLAG	Numerical expression of how alignment to the reference sequence occurs (bitwise).
RNAME	Reference sequence name. Generally, the chromosome name is used.
POS	The position of the first base on which the read is aligned on the reference sequence.
MAPQ	Mapping quality.
CIGAR	Compact Idiosyncratic Gapped Alignment Report. In the CIGAR region, “M” represents mismatch, “I” represents the insertion of more than one base, “D” represents deletion of more than one base.
RNEXT	For paired reads, the reference sequence name to which the next read is aligned. It is the RNAME equivalent for the next sequence.
PNEXT	The position at which the next read is aligned on the reference sequence.
TLEN	Template (DNA fragment) length.
SEQ	Segment sequence.
QUAL	ASCII of phred-scaled base quality.

To increase the performance in the analysis and to reduce memory usage, the analysis continues with the binary alignment map (BAM) extension, which is the compressed binary form of the SAM file [Li et al., 2009].

Reads can be aligned to the reference genome or to the reference transcriptome during the alignment of RNA data. While it is more advantageous to align to the

transcriptome to speed up the analysis, the disadvantage is that de novo transcripts cannot be detected.

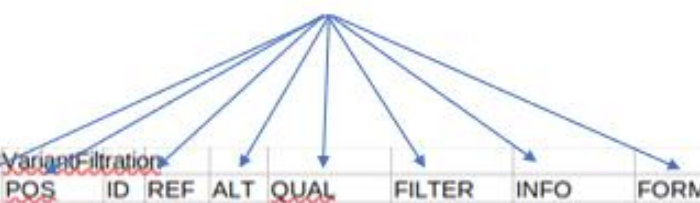
3.1. Variant Calling from RNA-Seq Data

Variants are the difference from the reference genome in one or more bases. It can have different names according to the rate of occurrence in the population. Variants with a population frequency below 1% are called mutations, and variants with a population frequency above 1% are called polymorphisms [Karki et al., 2015]. Variants are divided into two main classes, single nucleotide variation (SNV) and insertion/deletion (INDEL) or structural variants such as copy number variations and fusion genes. A single base change is called a single nucleotide variant, while the addition of more than one base is called insertion, and the deletion of more than one base is called deletion. In the literature, insertion and deletion events are mostly expressed as INDEL.

The process of detecting variants from NGS data is variant calling. Different algorithms have been developed for somatic and germline variant calling. Germline variants occur in the germ cells and that are passed onto offspring. The germline variant affects all body cells, so it can be detected from any tissue. Somatic variants occur in somatic cells and are non-inherited. They are detected from the target tissue. The germline variants are detected easier than somatic variants. One of the major problems of somatic variant detection is caused by tumor heterogeneity, which is a result of low variant frequency [Neums et al., 2018]. There are many tools developed to detect germline and somatic variants, including Genome Analysis Toolkit (GATK) [Auwera et al., 2013], VarScan [Koboldt et al., 2009], and SAMtools [Li et al., 2009b]. The output of the tools is called variant call format (VCF) file. VCF file consist of a header section and variant call records. The first eight columns in Table 3.2 are mandatory for the record section. Description in the Table 3.2 acquired from Samtools Github website [The Variant Call Format (VCF) Version 4.1 Specification, 2021.]. The definitions of the columns are in the header section. Filter column is empty before the variant filtration step. The filtration step is carried out to filter out false positives. The filtering criteria depends on the data and sequencing technology. And filtering is

done according to the quality metrics such as fisher strand, sequencing depth in the INFO section. Figure 3.5 shows the VCF file obtained after the variant filtration step.

VCF file columns



##source=VariantFiltration	CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT
chr1	14464	A	T	211.64	PASS	AC=1;AF=	GT:AD:DP:GQ:PL	0/1:8,7:15:99	
chr1	14522	G	A	159.64	PASS	AC=1;AF=	GT:AD:DP:GQ:PL	0/1:7,8:15:68	
chr1	14542	A	G	151.64	PASS	AC=1;AF=	GT:AD:DP:GQ:PL	0/1:4,6:10:86	
chr1	14574	A	G	29.64	FAIL	AC=1;AF=	GT:AD:DP:GQ:PL	0/1:14,5:19:37	
chr1	14653	C	T	195.64	PASS	AC=1;AF=	GT:AD:DP:GQ:PL	0/1:27,9:36:99	

Figure 3.5: Columns in a variant call record.

Table 3.2: Description of columns in a vcf file.

Column	Name	Description
1	CHROM	Name of chromosome
2	POS	Position of the variation
3	ID	Identifier of the variation
4	REF	Reference base at the position
5	ALT	Alternate base at the position
6	QUAL	Quality scores of the alleles
7	FILTER	Indicating whether the variant passed the filters
8	INFO	A list of values describing the variation
9	FORMAT	A list of describing the samples
+	SAMPLES	Sample files used for variant calling

The variant analysis is performed for DNA-Seq data such as WGS and WES. Whole exome sequencing (WES) variant analysis is mostly used in clinical implementation. WES increased the diagnostic efficiency of Mendelian diseases and enabled the discovery of new genes. Nevertheless, the diagnostic yield of WES analysis for rare diseases remains at 50%. This is because of prioritization, and interpretation problems. Although whole genome sequencing (WGS) allows detection of all variants, the diagnosis rate in clinical practice is close to WES. Furthermore,

prioritizing and interpreting the total 3 million of variants obtained from human WGS data remains challenging [Yépez et al., 2022].

Identification of genomic variants from RNA-Seq data continues to be a challenge because of the intrinsic complexity. However, in recent studies, it has been shown that variant analysis can be performed from RNA-Seq data due to its cost-effectiveness compared to WGS and the detection of variants such as splice or intronic variants which cannot be detected by WES [Adetunji et al., 2019].

Within the scope of the thesis, variant detection from RNA-Seq data was performed according to GATK best practice [<https://gatk.broadinstitute.org/>]. In this workflow, after the alignment step, duplicates are marked. After this step an additional step, compared to WES variant analysis, is applied which is performed with the SplitNCigarReads tool. SplitNCigarReads tool is used when RNA reads are aligned against the reference genome. This tool splits reads that contain Ns in their cigar string (e.g., spanning splicing events in RNA-Seq data). The aim of this process is hard clipping any sequences overhanging to the intronic regions.

The workflow continues with variant calling and variant filtration after SplitNCigarReads. Variant filtration step is applied to filter false positive variants. Variant calling tools report all differences as variants. However, variants that pass certain criteria such as alignment quality, and strand bias are considered true positive variants. The filters recommended by GATK for RNA-Seq variant filtration are as follows: If the variant is found on only one thread in paired reads (only forward or reverse), this variant is considered false positive. The Fisher strand score represents this criterion, and variants with $FS > 30$ for RNA-Seq data were considered false positives. For the quality by depth (QD) criterion, which expresses variant reliability, variants with $QD < 2$ according to GATK are considered false positive.

3.1.1. Variant Annotation

Annotation is a process where information stored in databases is assigned to variants. After the annotation step, the variants are interpreted and prioritized with information such as the population frequency of the variant, its effect on protein function, etc. There are many annotation tools including Variant Effect Predictor (VEP) from ENSEMBL [McLaren et al., 2016], GATK Variant Annotator [McKenna et al., 2010], SnpEff [Cingolani et al., 2012]. Within the scope of this thesis, variant

annotation was performed with ANNOVAR [Wang et al., 2010], a widely used annotation tool.

The ANNOVAR tool utilizes databases from the UCSC Genome Browser or any annotation dataset conforming to Generic Feature Format version 3 (GFF3). Gene based, filter based, or region-based annotation are performed by the ANNOVAR.

Gene Based Annotation is used to identify the gene in which a variant is located, the gene region (exonic, intronic, UTR3, UTR5 etc.), its function on the protein (synonymous, non-synonymous), the affected transcript and the amino acid change. Users can annotate variants in the RefSeq genes, UCSC genes, ENSEMBL gene or GENCODE genes.

Region based annotation is used to identify the variants overlap with certain regions of interest such as cytogenetic bands, microRNA target sites.

Filter based annotation is used to identify variant population frequency utilizing databases such as the Genome Aggregation Database (gnomAD; (gnomAD) [https://gnomad.broadinstitute.org] or 1000 genome project [The 1000 Genomes Project Consortium, 2010], variant pathogenicity and in silico scores such as SIFT [Ng and Henikoff, 2003] and Polyphen2 [Ramensky, 2002].

The ANNOVAR databases used within the scope of the study are shown in Table 3.3 and Table 3.4. s

Table 3.3: Databases for gene based annotation in ANNOVAR.

Database	Feature	Information
ensgene	chr, position	Chromosome and variant position in the genome.
	ref allele, alt allele	Reference allele (Ref allele) identical and alternate allele (Alt allele) not to the reference genome.
	consequence	The functional consequences of the variant (Contains 11 functional consequences including synonymous, non-synonymous).
	gene region	Region of variant hits. (Exonic, intronic, UTR3, UTR5, upstream, downstream, intergenic, splicing, ncRNA).

Table 3.4: Databases for filter based annotation in ANNOVAR.

Database	Feature	Information
dbNSFP	SIFT	D: Deleterious (SIFT $\leq$ 0.05). T: tolerated (SIFT $>$ 0.05).
	PolyPhen HDIV	D: Probably damaging ($\geq$ 0.957). P: possibly damaging (0.453 $\leq$ pp2_hdiv $\leq$ 0.956). B: benign (pp2_hdiv $\leq$ 0.452).
	PolyPhen HVAR	D: Probably damaging ($\geq$ 0.909). P: possibly damaging (0.447 $\leq$ pp2_hdiv $\leq$ 0.909). B: benign (pp2_hdiv $\leq$ 0.446).
	LRT	D: Deleterious N: Neutral U: Unknown
	MCAP	D: Deleterious (score $>$ 0.025) T: Tolerated (score $\leq$ 0.025)
	Mutation Tester	A: Disease_causing_automatic. D: Disease_causing. N: Polymorphism. P: Polymorphism automatic
	Mutation Assessor	H: high. M: medium. L: low. N: neutral. H/M means functional, and L/N means non-functional.
	MetaLR	D: Deleterious; T: Tolerated
	MetaSVM	D: Deleterious; T: Tolerated
	PROVEAN	D: Deleterious; N: Neutral
	FATHMM	D: Deleterious; T: Tolerated
	fathmm_coding	D: Deleterious; T: Tolerated
	DEOGEN2	D: Deleterious; T: Tolerated
	ClinPred	D: Deleterious. T: Tolerated.
gnomAD	Variant Frequency	Allele frequency in gnomAD database WES and WGS data on multiple populations.

All in silico scores used in the study were obtained from the database named dbNSFP. These scores are specific to nonsynonymous variants. The descriptions of the scores are as follows:

SIFT (Sorting Intolerant from Tolerant): It is the score showing the effect of amino acid substitution on protein function. SIFT algorithm uses sequence homology

considering the position at which the change occurred and the type of amino acid change [Ng and Henikoff, 2003].

PolyPhen2 (Polymorphism and Phenotyping v2): PolyPhen2 is the score shows the effect of amino acid substitution on structure and function of human protein. While SIFT does not require structure, PolyPhen2 uses this information. It makes computations on 3D models of the protein structure. Two pairs of datasets were used while training Polyphen2. The first pair is HumDiv and the second pair is HumVar. HumDiv (PolyPhen2 HDIV) consists of damaging alleles with known effects on the molecular function causing human Mendelian diseases and human proteins that assumed non-damaging. HumVar consists of human disease-causing mutations from UniProt (“UniProt: The Universal Protein Knowledgebase,” 2017) and non-synonymous variants which were treated as non-damaging [Adzhubei et al., 2010].

LRT (Likelihood Ratio Test): LRT shows the effect of missense changes on the function of human protein. LRT identifies a subset of deleterious mutations using comparative genomics dataset of 32 vertebrate species [Chun and Fay, 2009].

MutationTaster: MutationTaster scores show the disease-causing potential of DNA sequence alterations [Schwarz et al., 2010].

Mutation Assessor: Mutation assessor algorithm uses a multiple sequence alignment (MSA). Scores show impact of a missense variant [Reva et al., 2011].

MetaLR and MetaSVM: These are used to evaluate other prediction scores using two ensemble-based approaches, support vector machine (SVM) and logistic regression (LR) [Dong et al., 2015].

PROVEAN (Protein Variation Effect Analyzer): PROVEAN predicts the functional effects of protein sequence variations including single or multiple amino acid substitutions, and in-frame insertions and deletions. PROVEAN consists of two main steps: The first step is the collection of homologous and distantly related sequences (with BLASTP) and the second step is clustering [Choi et al., 2012].

FATHMM (Functional Analysis Through Hidden Markov Models): It is used to predict the functional effects of protein missense variants [Shihab et al., 2013].

FATHMM-coding: It is based on a machine learning approach (FATHMM-MKL (Multiple Kernel Learning)) that integrates functional annotations from ENCODE with nucleotide-based sequence conservation measures and predicts the functional consequences of coding variants [Shihab et al., 2015]. Pathogenic dataset

was constructed using heritable germ-line mutations from the Human Gene Mutation Database.

DEOGEN2: DEOGEN2 integrates information such as the molecular effect of the variant, the relevance of the mutated gene, its known interactions, and the pathways in which it is involved. The Random Forest Classifier from scikit-learn is used in DEOGEN2 to predict the effect of single amino acid substitution on protein [Raimondi et al., 2017].

ClinPred: It identifies disease related nonsynonymous variants [Alirezaie et al., 2018].

3.2. Expression Analysis from RNA-Seq Data

RNA-Seq data is mostly used to measure gene or transcript level in sample and to detect differentially expressed genes between two or more conditions [Stupnikov et al., 2021]. The expressed gene refers to the active gene. In other words, a functional product was created from DNA. RNA-Seq expression analysis begins with quantification. Two different approaches can be implemented to quantify at the gene level. In the first approach, the number of reads belonging to a gene is determined by counting the reads that overlap with the gene location. In the second approach, the number of transcripts corresponding to the gene is calculated and then the number of reads per gene is determined.

There are two major quantification methods. Alignment free quantification methods quantify transcripts with the pseudo alignment in k-mer space. Salmon [Bray et al., 2016] and Kallisto [Bray et al., 2016] are the popular alignment free and transcript level quantification tools. Alignment based quantification is a method to quantify only reads aligned to reference genome or transcript. Reads that are aligned to two or more gene locations are ignored. Quantification tools such as FeatureCounts [Liao et al., 2014] or htseq-count [Anders et al., 2015] ignore reads overlapping multiple genes in their default operating mode and do not count them. These two tools are the most common used tools in the quantification step. These tools are mostly preferred because of they are compatible with the tools used in differential expression analysis without the need for any processing. Table 3.5 shows an example of the read count matrix containing the raw read counts of the genes obtained from the

quantification step. In this matrix, each row corresponds to a gene and each column corresponds to a sample.

Table 3.5: The part of raw count matrix.

Gene ID	Case 1	Case 2	Ctrl 1	Ctrl 2
ENSG00000223972	1	0	1	1
ENSG00000227232	62	69	63	65
ENSG00000268903	35	26	2	13
ENSG00000269981	34	25	0	23

Samples in different conditions

Gene read counts in the read count matrix are not normalized. Raw read counts cannot be used to compare expression levels between samples. These counts need to be normalized regarding factors such as gene length. Genes with longer lengths are expected to have more read counts. In addition, reading depth is another factor that affects read counts. As the reading depth increases, the number of aligned reads will also increase [Evans et al., 2018]. Therefore, comparison of raw counts of samples with different read depths will not be reliable. Different methods have been developed for the normalization of read counts. TPM (transcript per million), RPKM (reads per kilobase of transcript per million reads mapped) and FPKM (fragments per kilobase of transcript per million reads mapped) are the normalization methods based on gene length and depth. FPKM is very similar to RPKM. RPKM is used for single-end RNA-seq while FPKM is used for paired-end RNA-seq. The formulas of the normalization methods are shown in Formula 3.1 and Formula 3.2.

$$\text{FPKM or RPKM} = \frac{\text{Number of reads mapped to gene} \times 10^3 \times 10^6}{\text{Total number of mapped reads} \times \text{Gene length(bp)}} \quad (3.1)$$

$$\text{TPM} = \frac{A \times 10^6}{\sum(A)} \quad A = \frac{\text{Total reads mapped to gene} \times 10^3}{\text{Gene length(bp)}} \quad (3.2)$$

Unlike FPKM, TPM is first normalized to the gene length and then to the total number of mapped read(depth). As a result, the sum of all TPM values is the same in all samples. Therefore, TPM is a more suitable normalization method for comparison

between samples than FPKM. FPKM is a more suitable normalization method for within sample comparisons. While new normalization methods are being developed, public databases and websites still use TPM or RPKM/FPKM. There is no consensus yet on which normalization method is the best [Conesa et al., 2016; S. Zhao et al., 2020; Y. Zhao et al., 2021].

Differentially expressed genes (DEG) can be detected using packages such as DESeq. There are many tools that make bioinformatic analysis practical. DESeq2 [Love et al., 2014], EdgeR [Robinson et al., 2010] and limma [Ritchie et al., 2015] tools are among the packages coded in R and widely used in DEG analysis. These tools take the raw read counts as input and output the normalized expression values, statistical p values and fold change for each gene.

3.3. Splicing Analysis from RNA-Seq Data

Splicing is a process in which introns are removed from the pre-mRNA and exons are ligated, resulting in the transcript being converted into mature mRNA. Alternative splicing is the process of ligating different exon combinations to transform the final mature mRNA. It is possible to produce more than one protein from one gene with alternative splicing processes. In Figure 3.6, where common alternative splicing events are shown, black lines represent introns and blue, orange and pink boxes represent exons. In constitutive splicing, introns are removed, and exons are ligated. In exon skipping, cutting occurs from the receiving region of the next exon instead of successive exons, and introns are removed along with the exon in between. The remaining exons are ligated. Alternative 5' splice site is upstream donor site and intron, with a part of exon are removed. The alternative 3' splice site is downstream of the acceptor site and intron, with a part of exon are removed. Intron retention is an alternative splicing event in which mature mRNA containing intron is occurred. Intron often contain premature termination codons, intron retaining isoforms are often degraded. If intron retaining isoforms are not degraded and converted to truncated protein isoforms. Splicing process can be repeat, if intron retaining isoforms are retained in the nucleus or cytoplasm [Zheng et. al, 2020].

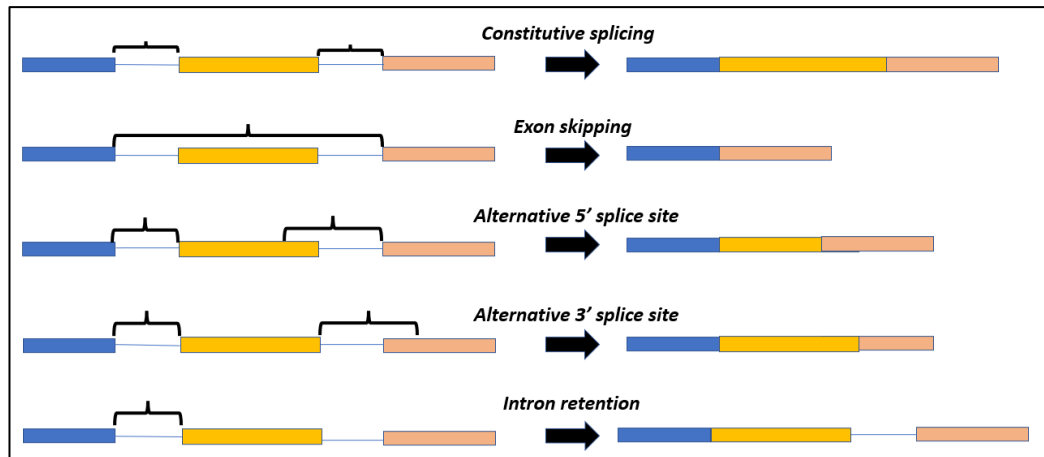


Figure 3.6: Common alternative splicing events.

The removal of the intron is carried out by the ribonucleoprotein complex called the spliceosome [Clancy, 2008]. Splicing is regulated trans-acting proteins (repressors and activators) and cis-acting regulatory sites (enhancer and silencer). When the repressor proteins bind to the silencer site, the splicing process is inactivated. When the activator proteins bind to the enhancer site, the splicing process is activated [Wang et al., 2015]. The boundaries between introns and exons are called splice sites. Upstream part of the intron is called the donor splice site and downstream part of the introns is called the acceptor splice site. As shown in Figure 3.7, the first two nucleotides GT (donor splice site) at the beginning of the intron and the last two nucleotides AG (acceptor splice site) represent the canonical splice site [Baten et al., 2006]. 98.7% of splice site pairs are canonical [Anna and Monika, 2018]. Splice site pairs, AT-AC and GC and AC, respectively, are called non-canonical. The canonical and non-canonical splice sites are known for their different snRNP combinations [Baten et al., 2006].

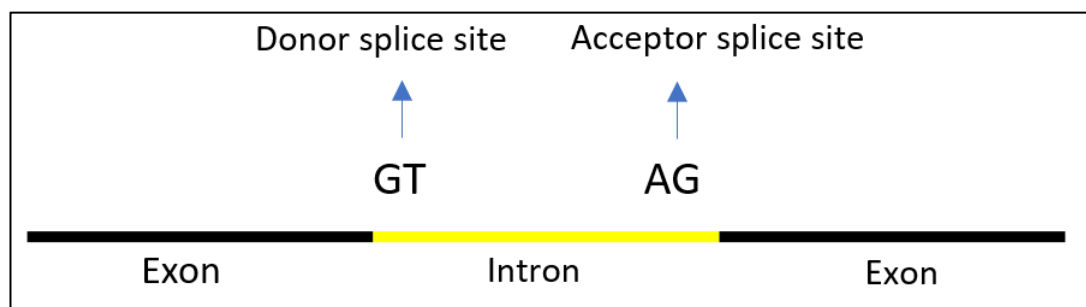


Figure 3.7: Canonical splice site.

A variant that occurs at the boundary of an exon and an intron (splice site) is called splice site mutation. 15-30% of variants causing hereditary diseases are splice site mutations [Anna and Monika, 2018]. Furthermore, about 62% of all pathogenic single nucleotide variants are thought to affect RNA splicing [Wai et al., 2020]. It is known that variants occurring in non-coding regions may cause aberrant splicing. Studies in the literature have also shown that RNA-Seq splicing analysis contributes to the interpretation and prioritization of variants and increases diagnosis rate [Marco-Puche et al., 2019; Curry et al., 2021].

There is no best practice for splicing analysis. However, there are several specialized tools for predicting splice changes. Early published splicing tools such as MaxEntScan focus on the close neighbourhood of splicing junctions. In the last few years, the number of published artificial intelligence based splicing tools has increased with the use of machine learning for biological data. Recently, deep neural networks (DNNs) have also achieved good results in predicting genome-wide splice variants. SPANR/SPIDEX is the first deep learning (DL) based tool. SPANR/SPIDEX [Xiong et al., 2015] model trained with the experimentally observed exon skipping events [Rentzsch et al., 2021].

Within the scope of the thesis, splicing analysis was performed with the SpliceAI [Jaganathan et al., 2019] tool released by Illumina. SpliceAI is another DL based tool. After the release of the SpliceAI tool, tools named FRASER [Mertes et al., 2021] and SCISSOR [Choi et al., 2021] were published that statistically detect aberrant splicing. However, these tools do not provide variant level information. In addition, enough control samples are required to use the FRASER tool. Therefore, these tools were not preferred for the splicing analysis.

SpliceAI analyses each position in a pre-mRNA transcript and evaluates whether it is likely to be a splice donor, acceptor, or neither. The model was only trained on reference transcript sequences and splice junction annotations, and never saw variant data during training. Training set was constructed with protein-coding transcripts from the GENCODE v24 annotation. The effect of the variant on splicing is evaluated according to the delta score produced by the tool. The tool predicts firstly exon-intron boundaries for both the reference pre-mRNA transcript sequence and the alternative transcript sequence containing the variant and then takes the difference between the scores [Jaganathan et al., 2019].

The SpliceAI tool uses a .vcf file, a reference genome, and an annotation file (.txt) as input. Annotation file contains chromosome, strand, transcript start (TX_START) and end (TX_END) position and exon start end positions. The tool cannot calculate for genes that are not included in the annotation file in the tool. The file should be updated by adding the relevant information of the genes for which calculation is requested to this file, as in Figure 3.8. The tool generates a vcf file that includes the SpliceAI scores as output.

#NAME	CHROM	STRAND	TX_START	TX_END	EXON_START	EXON_END
OR4F5	1	+	69090	70008	69090,	70008,
OR4F16	1	-	685715	686654	685715,	686654,
SAMD11	1	+	925737	944575	925737,925921,	925800,926013,930336,931089,93589
NOC2L	1	-	944203	959290	944203,945056,	944800,945146,945653,946286,94654
KLHL17	1	+	960586	965715	960586,961292,	960800,961552,961750,962047,96247
PLEKHN1	1	+	966496	975108	966496,966703,	966614,966803,970423,970601,97075
PERM1	1	-	975204	981029	975204,976498,	976269,976624,981029,
HES4	1	-	998963	1000111	998963,999525,	999432,999613,999787,1000111,
ISG15	1	+	1013422	1014540	1013422,101398,	1013576,1014540,
AGRN	1	+	1020122	1056116	1020122,102220,	1020373,1022462,1035324,1040880,
RNF223	1	-	1070965	1074307	1070965,107401,	1072575,1074307,
C1orf159	1	-	1081822	1116098	1081822,108435,	1082987,1084383,1084506,1086012,
TTLL10	1	+	1173902	1197935	1173902,117426,	1173926,1174321,1174489,1179333,
TNFRSF18	1	-	1203507	1206571	1203507,120439,	1203960,1204486,1205492,1206571,
TNFRSF4	1	-	1211325	1214132	1211325,121170,	1211625,1211832,1212138,1212704,
SDF4	1	-	1216930	1232031	1216930,121845,	1217804,1218633,1218927,1223357,
B3GALT6	1	+	1232264	1235041	1232264,	1235041,
FAM132A	1	-	1242452	1246722	1242452,124283,	1242646,1242913,1243152,1243552,
UBE2J2	1	-	1254469	1273885	1254469,125604,	1255487,1256125,1257130,1257310,

Figure 3.8: Part of the annotation file.

The output file contains eight values for each variant, of which four are delta scores and four are positions. the scores show the effect of the variant on splicing, while the positions indicate the position where the splicing changes relative to the variant position.

4. ML BASED STRATEGIES FOR CLINICAL CHARACTERIZATION OF GERMLINE VARIANTS

The American College of Medical Genetics and Genomics (ACMG) collaboration with the Association of Molecular Pathology (AMP) have published a guideline consisting of 28 criteria to be used for interpretation of germline variants in 2015. The criteria are grouped according to biological impact, in silico predictions, presence in the control cohort, familial information, and inheritance mode. Variants are evaluated hierarchically and divided into 5 main classes as pathogenic, likely pathogenic, benign, likely benign and VUS according to the proposed criteria [Richards et al., 2015]. Manual evaluation of these criteria one by one is challenging as it is time consuming and requires expertise. Several tools have been developed that automate the ACMG criteria but are not machine learning based [Li and Wang, 2017; Ravichandran et al., 2019]. However, these tools are limited due to their gene specific nature.

ML based tools developed in line with the ACMG criteria and published in the literature to automate the variant classification with additional features [Lai et al., 2020; Nicora et al., 2018]. Since some of these criteria evaluate variants according to familial information, not all criteria are suitable for general use. Generalizable criteria such as variant population frequency from ACMG-AMP criteria were used as features in ML based tools. The success of ML-based methods depends on the selected training dataset and features. The features used in the models developed so far can be obtained from databases or can be calculated by DNA-Seq-based bioinformatic analysis methods. These methods are still insufficient for characterizing remaining variants in the deep intronic region. Therefore, there is a need for a tool that uses features obtained from RNA-Seq data.

4.1. Machine Learning Based Approaches

Artificial intelligence is a technique in which functions such as learning and problem solving that require human intelligence are performed through computer systems. Machine learning is a subarea of artificial intelligence. In machine learning,

algorithms that can learn the connections in input data by themselves and without human intervention are developed.

Machine learning algorithms are used to extract useful patterns from complex data by utilizing experience. For this purpose, it uses various statistical, probabilistic and optimization methods. Machine learning algorithms can be generally divided into four main categories, namely supervised, unsupervised, semi-supervised and reinforcement learning. In supervised machine learning, algorithms produce a function that maps inputs (data) to desired outputs (label). Training data consists of both inputs and outputs. Classification and regression tasks are usually supervised machine learning algorithms. Classification is the task of predicting the class label, such as male/female, true/false, while regression is the task of predicting a continuous quantity, such as a prediction of house prices or the weather forecast.

In unsupervised algorithms, it is unclear which class the input data belongs to. These algorithms utilize a function to extract an unknown structure over unlabelled data. Clustering and association are examples of tasks that can be accomplished by unsupervised machine learning algorithms. Clustering is the process of assigning similar observations to the same clusters. An association algorithm analyses the co-occurrence of events and establishes relationships between data.

Semi-supervised learning is a method that falls between supervised and unsupervised learning. These types of algorithms are used for improving the accuracy of learning generally when there are small amount of labelled data for training and a large amount of unlabelled data [Edwards, 2018.]. Reinforcement learning is a method that enables an agent to learn through trial and error in an interactive environment using feedback from their own actions and experiences. Reinforcement learning mostly used to train robot motions.

In machine learning, the input is known as the instance (data) and the output is known as the target (label). The input used for training is a series of data called set of features. The features can either be continuous, binary, or categorical. Categorical features must be converted to numeric due to machine learning models require all input and output variables to be numeric. The terms, validation data and test data are frequently handled along with training data. The validation data is used to set the model parameters correctly during model training. Validation data is not used in unsupervised methods because it is not strictly correct. The test data is the input that

the model does not encounter during the training. Model performance is estimated using the test data.

Data and model fit are very important for the success of the model. Compared to the data complexity, very complex or simple models would reveal overfitting or underfitting problems. As shown in Figure 4.1, the model selection process requires a trade-off between bias and variance (Figure 4.1 is taken from Towards Data Science website [Mottaghinejad, 2021]). The bias is the value that reflects the distance between the estimated data and the actual data because of the modelling. The variance is the variability of the model estimate for a given data point, or the value that indicates how the data is spread out. High bias and low variance are interpreted as underfitting. In other words, the model is simplistic when it is compared to the data complexity. In this case, the accuracy is low since the model cannot learn the data sufficiently. Low bias and high variance are interpreted as overfitting. The model is more complex than needed. The model picks up important relationships between variables, as well as relationships that turn out to be just the result of a noise. Hence the model cannot perform well on the test data because it memorizes the training data.

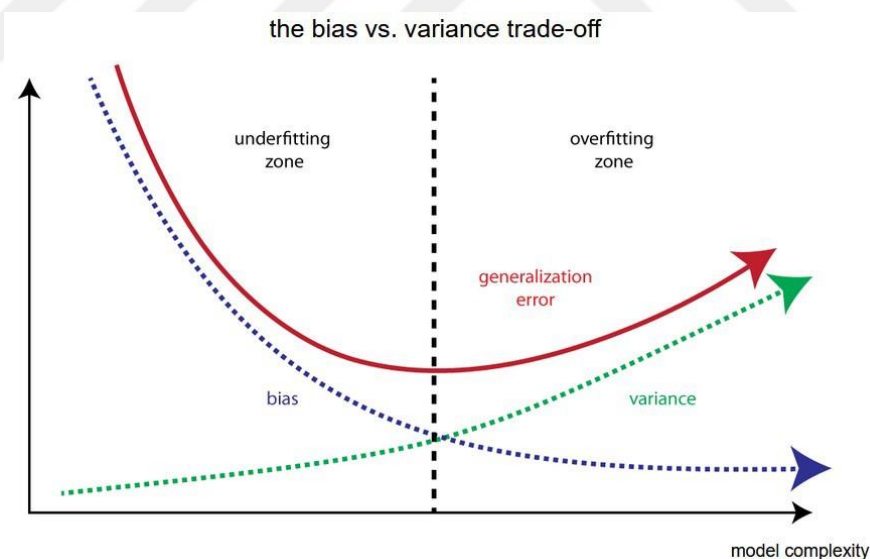


Figure 4.1: Model selection in machine learning.

4.2. Random Forest Model

Random Forest is a supervised machine learning model developed by Breiman [Breiman, 2001]. The random forest algorithm is built on decision tree models. Figure

4.2 shows an example of a decision tree. The nodes with the first split and representing the entire sample is called the root node. If a sub-node is divided into further sub-nodes, it is called a decision node. Nodes that represent the class or label that are not divided into further sub-nodes are called leaves or terminal nodes. Each node in the tree represents some features edge represent possible answer.

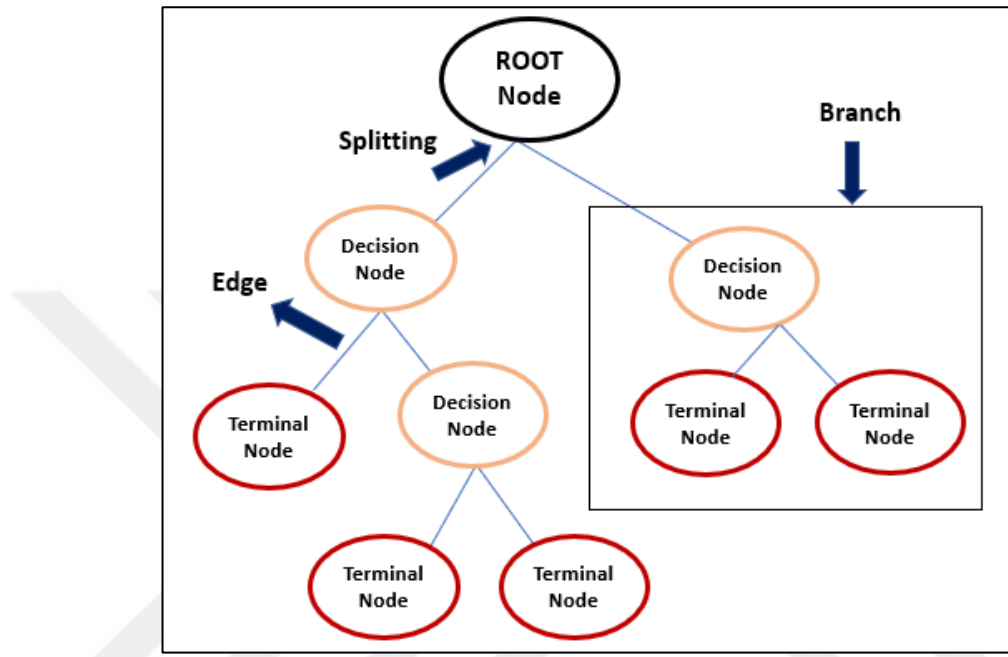


Figure 4.2: An example of decision tree.

The purpose of dividing into sub nodes is to increase homogeneity. The purest node or most homogeneous node has samples from only one class. Node splitting is one of the factors affecting the performance of decision trees. There are different node splitting strategies depending on whether the target variable is categorical or continuous. If the target variable is categorical (such as classification problem), information gain or entropy is used, otherwise (such as regression problem) reduction in variance is used. The variance is calculated to decide the homogeneity of the node in reduction in variance. A completely homogeneous node has zero variance.

Entropy can be defined as a measure of the disorder and randomness. The lower the entropy, the higher the purity in node splitting. The range of Entropy lies in between 0 to 1. The entropy formula is shown in Formula 4.1, where p represents class probabilities and n represents all classes [Hershy, 2019].

$$\text{Entropy} = - \sum_{i=1}^n p_i \log_2 p_i \quad (4.1)$$

Information gain is calculated using the difference in entropy before and after a split and measures purity. The purpose of information gain is to identify the most important features that contain the most information about a class.

In random forest, most popular splitting criterion is gini impurity. The gini property measures the impurity of a node. If all the training samples in the node are of the same class, the node is pure (gini = 0). While gini's maximum impurity is 0.5 and the entropy's is 1 [Hershy, 2019]. The gini formula is shown in Formula 4.2.

$$\text{Gini} = 1 - \sum_{i=1}^n p_i^2 \quad (4.2)$$

Random forest consists of many decision trees. As shown in Figure 4.3, The predictions from individual trees are collected, and the final class is estimated by combining the individual predictions using a method such as the majority voting for classification and averaging for regression.

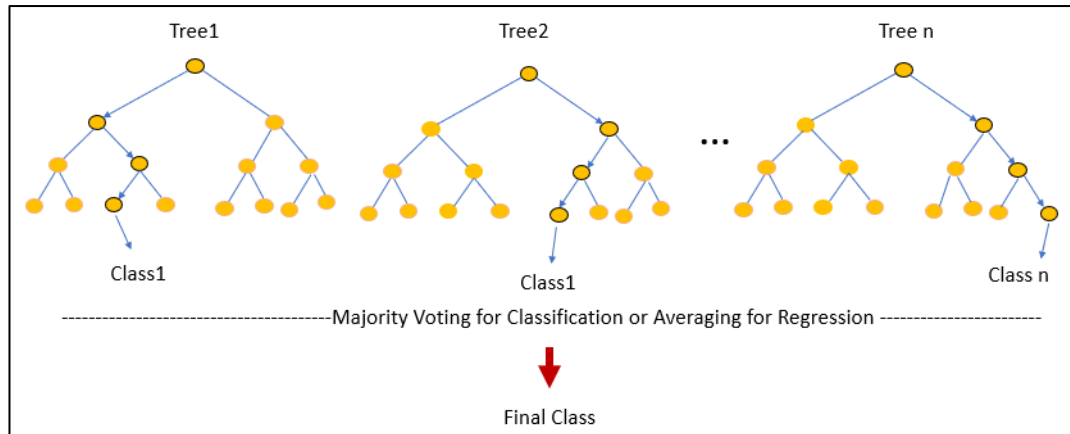


Figure 4.3: An example of a random forest.

In fact, each prediction is the output of a model. This process of combining different model results is called ensemble learning. The ensemble learning method used by the random forest algorithm for training is bagging or bootstrap aggregation. The concepts of using multiple trees in parallel and bootstrapping results in

overcoming of overfitting issues by random forests [Qi, 2012]. Bootstrapping is the selection of random samples from the dataset with replacement.

4.3. Model Performance Metrics

Model performance is calculated according to the differences between the correct prediction by the model and the actual class in supervised machine learning algorithms. The closer the model predicts the actual class, the higher the performance of the model. There are different performance metrics that are used for model evaluation purposes. To calculate these metrics, the numbers of true positive (TP), false positive (FP), true negative (TN), and false negative (FN) predictions are employed. In the confusion matrix shown in Figure 4.4, the actual classes are assumed to be positive and negative. It is called TP, when the model predicts a member of positive class as positive, and FN when it predicts the member as negative. It is called TN when the model predicts a member of negative class as negative, and FP when it predicts the member as positive.

		PREDICTED CLASS		
		Positive	Negative	
ACTUAL CLASS	Positive	TP	FP	Sensitivity / Recall $\frac{TP}{(TP + FN)}$
	Negative	FN	TN	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Values $\frac{TN}{(TN + FN)}$	Performance metrics

Figure 4.4: Confusion matrix and performance metrics.

The performance of classification problems can be evaluated using accuracy, F1 score, sensitivity, or specificity. Accuracy represents the number of correctly classified

samples over the total number of data samples. Accuracy considers the number of all correctly classified samples and do not concentrate on a specific class. Accuracy calculation is demonstrated in Formula 4.3.

$$\text{Accuracy} = \frac{TP + TN}{(TP + TN + FP + FN)} \quad (4.3)$$

Since there are different numbers of negative and positive samples in the imbalanced data, the F1 score is a better measure to evaluate model performance instead of accuracy. F1 score calculation is demonstrated in Formula 4.4. If the positive class mattered the most as compared to negative, both precision and recall are crucial for the evaluation of model performance. F1 score is balancing precision and recall on the positive class while accuracy looks at correctly classified observations of both positive and negative set.

$$\text{F1 score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (4.4)$$

Sensitivity is the positive class detection performance of the model, that is, it shows how many of the members in the positive class are correctly classified. Specificity, the opposite of sensitivity, is the negative class detection performance of the model, that is, it shows how much of it is classified as negative within the negative class. The formulas of sensitivity and specificity are shown in Figure 4.4.

5. MODEL AND DATASET CONSTRUCTION

Random forest is a popular supervised machine learning algorithm that can be used for both regression and classification problems. It is an easy-to-use algorithm that gives fast results. In addition, it can give good results even without extensive parameter setting process. Therefore, as in most variant classifier tools in the literature [Alirezaie et al. 2018; Lai et al. 2020], the random forest classifier was used while developing the clinical characterization model of germline variants within the scope of the thesis. The scikit-learn python library was preferred because it facilitates the calculation of feature importance, as well. It automatically calculates the contribution of each feature to build decision trees during training.

The feature table which is input to the random forest classifier model was prepared as follows: First, public RNA-Seq data was collected to be used in bioinformatic analyses. All studies on mendelian or hereditary cancer with published raw RNA-Seq data were considered, during data collection. The publicly available datasets of the studies were obtained from GEO database [Edgar et al., 2002]. The datasets consisted of patients (with mendelian and hereditary cancers) and control individuals. Table 5.1 shows the accession IDs and disease types of these datasets.

Table 5.1: Accession IDs and diseases.

Accession ID	Disease Type	Number of Case Samples	Number of Control Samples	Tissue
GSE85984	Myotonic dystrophy of type 1 (DM1)	3	3	Skeletal muscle
GSE157428	Myotonic dystrophy of type 1 (DM1)	21	8	Frontal cortex
GSE136371	Cystic Fibrosis	33	16	Whole blood
GSE106500	Lynch Syndrome / Familial Adenomatous Polyposis	11/13	0	Colon
GSE149276	Breast Cancer	34	0	Breast

Myotonic dystrophy type 1 (DM1) is an inherited muscle disease that causes progressive muscle weakness and loss. When one copy of the gene is altered, the disease appears in the phenotype. This indicates that the inheritance pattern is autosomal dominant. [Meola and Cardani, 2015]. Cystic fibrosis (CF) is an important genetic disease that affects various biological systems, especially the lungs and digestive system. Inheritance pattern of CF is autosomal recessive. Mutations occur in both copies of the gene in autosomal recessive diseases. [Davies et al., 2007]. CF affects cells that produce mucus, sweat, and digestive juices. Lynch Syndrome (LS) and Familial Adenomatous Polyposis (FAP) are autosomal dominant (dominant) diseases. The most common hereditary colon cancer is non-polyposis colorectal cancer. This disease is also known as Hereditary Non-polyposis Colorectal cancer (HNPCC) or Lynch Syndrome. Unlike familial adenomatous polyposis (FAP) syndrome, hundreds of polyps do not occur in these patients [Lynch et al., 2009]. Breast cancer is a type of cancer that is very common in women. 70% of breast cancer is sporadic, 10% is hereditary and 20% is familial. Hereditary breast cancer exhibits dominant autosomal inheritance [Haddad, 2020].

Paired RNA-Seq data collected from samples with disease (case samples) in the datasets were used for variant calling, variant annotation, expression, and splicing analyses, respectively. The goal of variant calling analysis is to detect germline variants related to the hereditary diseases. For this purpose, HaplotypeCaller [Auwera et al., 2013b] was used in the variant calling step and the variant list was obtained in vcf format.

Databases downloaded from ANNOVAR [Wang et al., 2010] tool (2020) were used for variant annotation. Databases in ANNOVAR are explained in Section 3.1.1. Accordingly, the databases used are as follows: gnomAD (2019) for the population frequency, dbNSFP (v.42a) for the in-silico prediction scores, ClinVar (2022) and InterVar (2018) databases for the clinical characterization of the variants. The InterVar database classifies variants according to ACMG-AMP criteria. This classification is used together with the ClinVar database for variant characterization.

The ClinVar database does not divide variants into 5 main classes as pathogenic, likely pathogenic, benign, likely benign, and VUS, as in the ACMG criteria. ClinVar characterizes variants in different terms, including these main classes [<https://www.ncbi.nlm.nih.gov/clinvar/docs/clinsig/>]. Disagreement among these

terms is the 'Conflicting Interpretation of Pathogenicity'. Conflicting Interpretation of Pathogenicity variant label represents the variants reported differently by different submitters of ClinVar database. For example, one such variant may be reported as possibly pathogenic by one submitter, while another submitter may report the same variant as likely benign. Since there is no common decision on such variants, they were ignored in this study and treated like VUS.

The pathogenic and benign variants were selected as follows: If a variant is pathogenic/likely pathogenic according to both databases (ClinVar and InterVar) or pathogenic/likely pathogenic according to one database and not benign/likely benign according to the other database, it is included in the training set. The opposite has been done for benign ones. As a result, a total of 7015 pathogenic and 1285930 benign variants were obtained.

After identifying the pathogenic and benign variants according to the ClinVar and InterVar databases, filters for population frequency and consequence features are applied. For the first filter on population frequency, the threshold value of 0.05 was used due to the most commonly applied threshold value is 0.05 [Ghosh et al., 2018; Goswami et al., 2021]. According to the BA1 criterion of ACMG-AMP benign class, variants are evaluated according to frequency > 0.05 . All pathogenic variants in the ClinVar database which contains more variants than the InterVar database were evaluated. Only five of the pathogenic variants in ClinVar had a population frequency greater than 0.05. Out of these five variants in ClinVar, only two were detected in the variant analysis performed in our study.

Variants that do not change the amino acid structure are called as synonymous variants. Synonymous variants may be pathogenic according to the literature [Zeng and Bromberg, 2019]. However, only two of the variants in our study dataset are both pathogenic and synonymous.

As a result, it was decided to exclude the synonymous variants (included non-synonymous, stopgain etc. variants) and the variants with a population frequency greater than 0.05 from our dataset. They were not included in the study to prevent these small number of variants from complicating the model construction. After filtering, 2382 pathogenic and 227205 benign variants remained.

Expression analysis was performed, as well, for all samples in the datasets. The normalized expression values (TPM) were obtained by using the StringTie tool [Pertea

et al., 2015]. FC (fold change) was calculated by using normalized expression values at the gene level to detect differentially expressed genes. In the datasets including the control group, FC was calculated with the expression values of case and control individuals. The file named "RNA GTEx tissue gene data" in the Human Protein Atlas were downloaded and the TPM values in the file were used for datasets without control individuals. GTEx tissue gene data, a tab-separated file, includes TPM values of each gene in 37 different tissues.

Splicing effects of the variants obtained in the variant calling step were determined with the SpliceAI tool [Jaganathan et al., 2019]. While on Central Processing Unit (CPU), the tool annotates an average of 4 thousand variants per hour it annotates 25000 variants on Graphical Processing Units (GPU). So, the tool was run on Nvidia GeForce to speed up the analysis.

SpliceAI tool produces four delta scores for acceptor and donor site as output. As a result of the splicing analysis, it was determined whether the splicing process influenced by the variant or not according to values of the delta scores. The recommended value of 0.5 was used as threshold [Jaganathan et al., 2019]. The variants with delta scores greater than 0.5 were determined as splice-affecting variants. Such variants are labelled with 'yes' while variants not affecting splicing processes were labelled as 'no'. Two example variants with and without splicing effect determined based on the 0.5 cut-off are shown in Table 5.2.

Table 5.2: SpliceAI delta scores and categorization of the scores.

Variant	DS.AG	DS.DG	DS.AL	DS.DL	Max Delta Score	Cutoff: 0.5
chr19:11120523:G:C	0	0	0	0.99	0.99	yes
chr1:55057360:A:G	0	0.02	0	0	0.02	no

A feature table was prepared with the information obtained by bioinformatic analyses. The class labels should be given before training step due to the random forest model used in this thesis is a supervised machine learning algorithm. Accordingly, variants in the feature table were labelled as pathogenic or benign. Pathogenic and likely pathogenic variants were considered to be pathogenic and benign and likely benign variants were considered to be benign. Benign variants labelled with "0" in

datasets representing the positive class, and pathogenic variants labelled with "1" representing the negative class. Table 5.3 shows an example feature table for two variants.

A total of 2382 variants were labelled as pathogenic according to the ClinVar and InterVar databases. Since the number of benign variants is very big compared to pathogenic ones, benign variants in datasets were randomly selected. Two datasets with different number of benign variants were prepared. The first dataset is balanced so, as many benign variants as the number of pathogenic variants were selected. Path1xbenign1x tag is used for the balanced dataset in this thesis. In the second dataset, the number of benign variants selected were 10 times larger than the number of pathogenic variants. Path1xbenign10x tag is used for this imbalanced dataset in this thesis. As a result, the first dataset consists of 4764 variants; 2382 pathogenic and 2382 benign, and the second dataset consists of 26202 variants; 2382 pathogenic and 23820 benign variants.

Table 5.3: An example feature table.

	Variant 1	Variant 2
Label	1	0
Features		
Chr	3	19
Position	182872530	54221884
Ref	G	T
Alt	G	A
Gene Region	exonic	exonic
Genotype	Het	Het
Consequence	stopgain	nonsynonymous SNV
Population frequency		0.0469
FC	3.16	0.65
cat_spliceai	no	no
SIFT_pred		D
PolyPhen HDIV_pred		B
PolyPhen HVAR_pred		B
LRT_pred	D	N
MCAP_pred		
Mutation Tester_pred	A	N
Mutation Assessor_pred		N
MetaLR_pred		T

Table 5.3: Continuation.

MetaSVM_pred		T
PROVEAN_pred		N
FATHMM_pred		N
fathmm_coding_pred	D	T
DEOGEN2_pred		
ClinPred_pred		T
SIFT_score		0.027
PolyPhen HDIV_score		0.142
PolyPhen HVAR_score		0.216
LRT_score	0	0.071
MCAP_score		
Mutation Tester_score	1	1
Mutation Assessor_score		0.765
MetaLR_score		0.03
MetaSVM_score		-1.049
PROVEAN_score		-1.9
FATHMM_score		2.55
FATHMM_coding_score	0.995	0.006
variant_type	SNP	SNP

The performance evaluation of the models generated in this study is performed using both hold-out and cross validation tests. In cross validation, training set and test set are selected randomly from the dataset. On the other hand, the hold-out test a set of variants are selected from the dataset to represent the real-world operation.

For the cross validation, StratifiedKFold in scikit-learn is used with K=5. During this test, the data is randomly divided into 5 parts. One of them is used as test data and the rest are used for training. The training and test process is repeated 5 times where a separate part is used as test set.

For old-out tests, a test set consisting of variants that the model did not see during the training was prepared to evaluate the real-world performance of the model. So, this test set is called real-world dataset. As the real-world dataset, 2 case samples from the GSE157428 and GSE136371 datasets were used, with the sample IDs SRR12582120 and SRR10037663, respectively. There are 24 pathogenic variants in the sample SRR12582120, and 19 pathogenic variants in the sample SRR10037663. Other samples in these two datasets (GSE157428 and GSE136371) were used for model training. Therefore, there is a high probability of a common variants between other samples in the same dataset and the samples used for real-world dataset. Naturally, all the features of each common variant are same except the expression feature. If these

variants are included in the real-world dataset, with very high probability, they would be classified to be pathogenic since the model used quite similar variants during the training. This may result in artificially increased model performance on real-world dataset. Therefore, these common variants were not included in the real-world dataset to evaluate the performance of the model for characterizing completely diverse variants. After removing the common variants, 26 pathogenic variants remained in total, 8 from the sample SRR12582120 and 18 from the sample SRR10037663.

Since the number of benign variants is orders of magnitude larger than the number of pathogenic variants in real world samples, an imbalanced test set was prepared as real-world dataset by randomly selecting almost 10 times more benign variants. As a result, the real-world dataset consists of 326 variants, 26 pathogenic and 300 benign. All datasets used in the thesis are summarized in Table 5.4.

Table 5.4: Dataset and variant count.

Dataset	Number of Pathogenic Variants	Number of Benign Variants
path1xbenign1x	2382	2382
path1xbenign10x	2382	23820
real-world (hold-out)	26	300

As explained above, using the public RNA-Seq data, the features required for variant characterization were obtained via bioinformatic analyses and the datasets are formed. Figure 5.1 summarizes the dataset construction process.

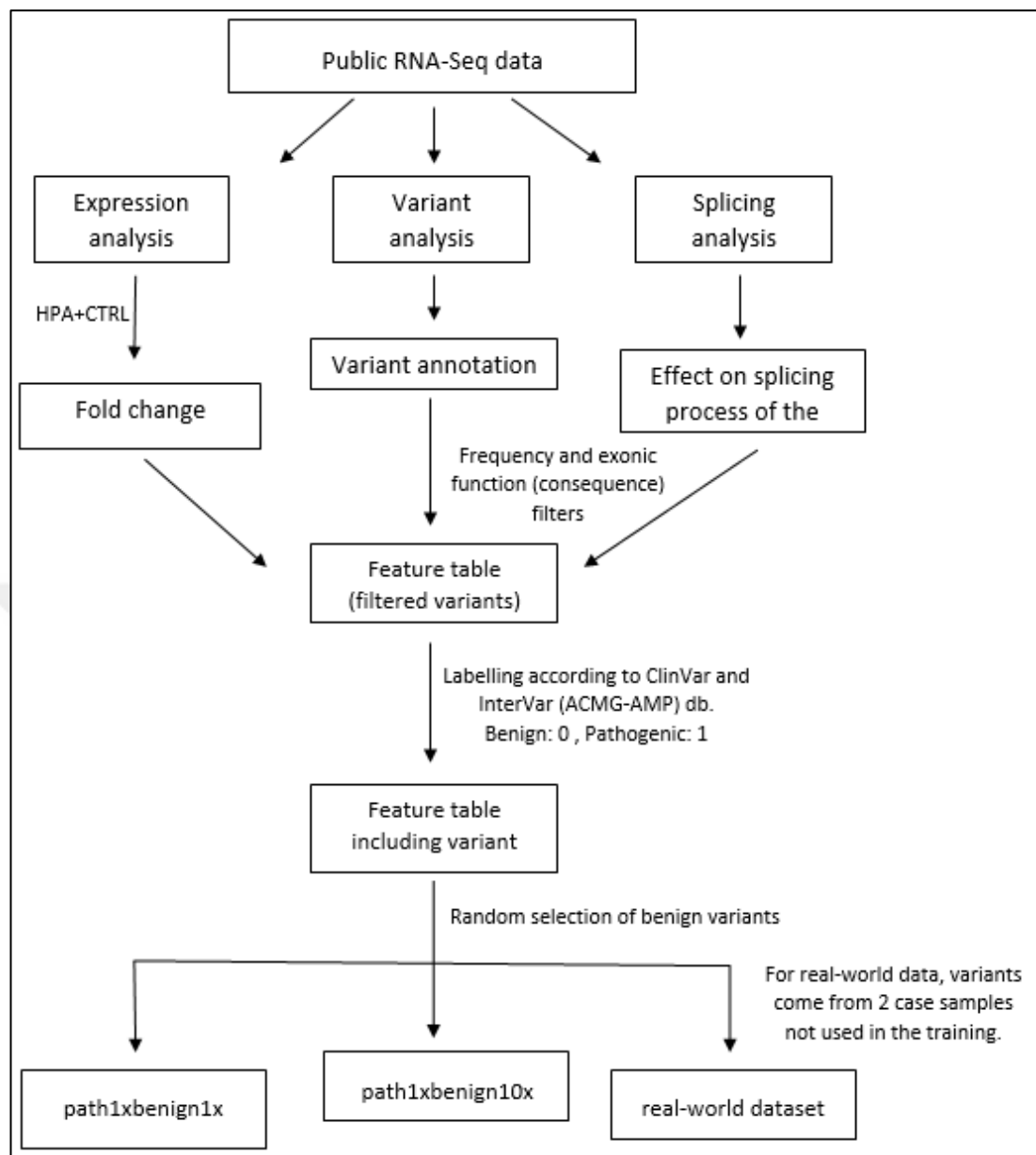


Figure 5.1: Summary of data construction step.

6. FEATURE ENGINEERING

In the datasets, the feature table contains both numeric (e.g., SIFT score and population frequency) and categorical data (e.g., Gene Region and Consequence). Also, some variants do not have value (missing value or NA) for a number of features (see Table 5.3). For example, there are variants without population frequency information. So, the features in the input data needs to be converted into a format suitable for the model. This process is a part of feature engineering in the literature [Rawat and Khemchandani, 2017].

Missing data can be handled by deleting or imputing. Deleting missing values will reduce the size of the dataset. Our main goal is to accurately detect pathogenic variants. Therefore, variants with missing information were not deleted so as not to reduce the number of pathogenic variants. Also, since features will be prioritized by the algorithm, the missing features for most variants were not deleted. Instead, missing values were imputed in our study. There are various techniques in statistics such as mean imputation and median imputation to impute missing data. In mean or median imputation, missing values are replaced with the mean or median of the values for the corresponding feature. However, this kind of imputation is not reliable for biological features.

The numerical data in the feature table are FC (expression feature), population frequency, position and in silico prediction scores. Except for the position, other numeric features contain missing data (NAs).

Three solutions are considered to solve missing value problem in FC feature. For the first two solutions, the FC values are categorized, and the category named NA is used to represent missing values. In the first solution, the cut-off used during conversion to categorical values are 1.5 and 0.66, which are frequently preferred in DEG analyses. As shown in the column named cat_FC1 in Table 6.1, if the FC value is greater than 1.5, the gene is up regulated and represented as "up". If the FC value is less than 0.6, the gene is downregulated and represented as "down". If the FC value is between 1.5 and 0.6, the expression of the gene is assumed to be unchanged and represented as "unchanged".

In the second solution, FC values including missing values were categorized into three classes. As shown in the column cat_FC2 in Table 6.1, if the genes are up-

regulated or down-regulated, they are represented as changed. If the gene expression does not change, it is represented as unchanged.

Table 6.1: FC and categorical expression values.

Variant	Gene	FC	cat_FC1	cat_FC2
chr13:32333046:A:G	BRCA2	11.67	up	changed
chr11:108267276:T:C	ATM	0.24	down	changed
chrX:154362534:G:A	FLNA	0.93	unchanged	unchanged
chr1:109236451:G:A	SARS	NA	NA	NA

In the third solution, numeric FC values were used directly for the expression feature, and NA's have been replaced with "-1" as a solution to the problem of missing value problem. The reason for not preferring a method such as mean or median imputation was that they are not reliable for biological features, as it is mentioned before. For example, if the mean is higher than 1.5, the model would learn that the variant has an effect on the gene expression. So, values such as mean or median for NAs can directly affect the decision of the model. The NAs were replaced with "-1" since the FC value can never take a negative value. Thus, this feature will be ineffective for variants without FC information.

Similar to FC values, the population frequency cannot be negative. Therefore, "-1" was used for variants without frequency information. Impact of these choices on the performance of the model is discussed in Section 7.

Replacing NAs with -1 is not applicable for in-silico prediction scores since these scores can have negative values, as well. So, they were categorized according to their letter scores. Letter scores were determined according to thresholds specified in the literature. For example, If the SIFT score is less than 0.05, it is denoted by the letter "D" representing "Deleterious", otherwise it is denoted by a "T" representing "Tolerated". The thresholds from the literature were used to categorize other in silico prediction scores. Furthermore, for the in-silico prediction scores of the dbNSFP database, ANNOVAR includes letter scores already converted according to the thresholds specified in literature. Therefore, these letter scores were obtained from the database without any manual transformation.

There are several categorical features in the data set; chr, ref allele, alt allele, gene region of the variants such as exonic or intronic, consequence showing the effect of the variant on the protein, variant type (SNP/INDEL), splicing effect, genotype, and letter scores of the in-silico prediction scores. These categorical features were converted to numeric by the “one hot encoding” method, which is a popular technique. In one hot encoding, binary representation of categorical variables is used. First, categorical values are mapped to integer values starting from zero. The binary vector representing a categorical value is obtained by setting values in the vector to zero except the value at the integer index which is marked with one. In this representation, missing data is another categorical value, and it is converted to numeric value similarly, so there is no need for an additional data imputation.



7. RESULT AND DISCUSSION

Within the scope of the thesis, RandomForestClassifier class of the sklearn.ensemble library was used in its default settings (if it is not specified otherwise explicitly) for model training. The number of trees (number of estimators) was 100 and class weight was none. In model training performed with imbalanced data class weight and minimum sample split parameters were experimented, as well. Cross validation tests were carried out with StratifiedKFold (K=5) method. Confusion matrix was used to show the performance of classification algorithm. The metrics used to evaluate the model performance were accuracy and F1 score. These scores, as well as sensitivity and specificity scores, were examined for the models trained with imbalanced data.

7.1. Feature Importance

A feature comparison was performed to observe the effect of the features calculated within the scope of the thesis on the classification models. The contribution of features is visualized as a column chart using RandomForestClassifier's gini importance method. In gini importance, the feature with a higher score has the more effect on classification.

In the dataset construction section, three different alternatives for representing the expression feature were presented. These alternatives, together with the representations for the other features were evaluated. All alternatives were added to the feature table separately and their effect on the classification was presented.

For the expression value, the effect of numeric and two categorization methods on the proposed model was examined. Feature importance charts are shown in Figure 7.1, 7.2 and 7.3 where each bar represents a feature. The feature in the yellow frame is the expression feature.

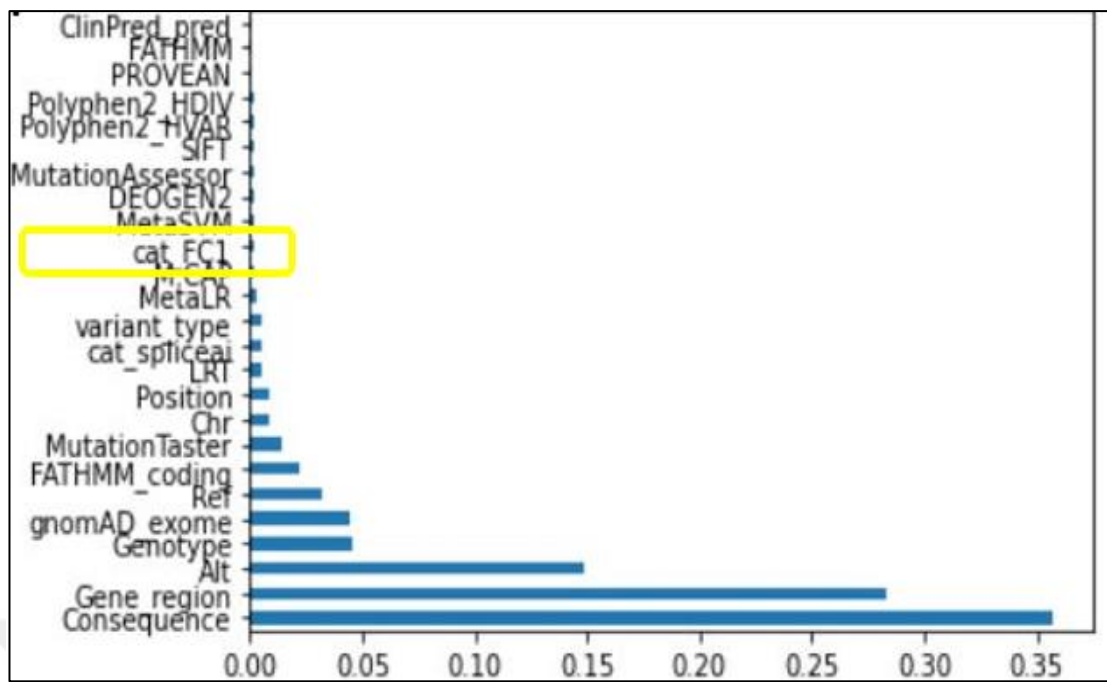


Figure 7.1: Feature importance charts for cat_FC1.

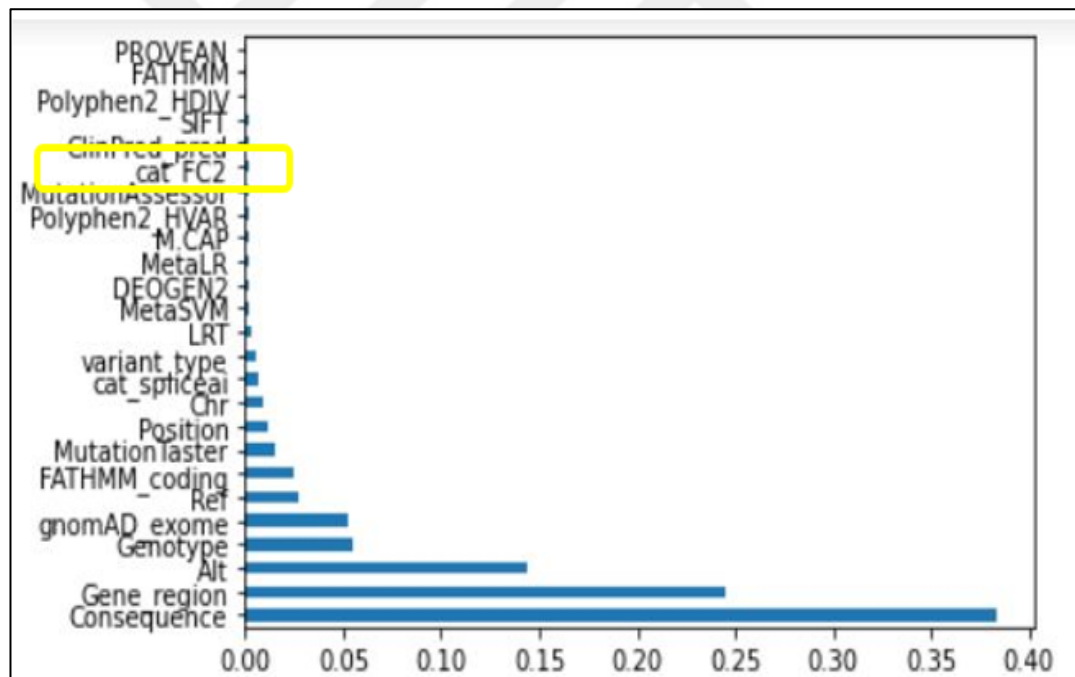


Figure 7.2: Feature importance charts for cat_FC2.

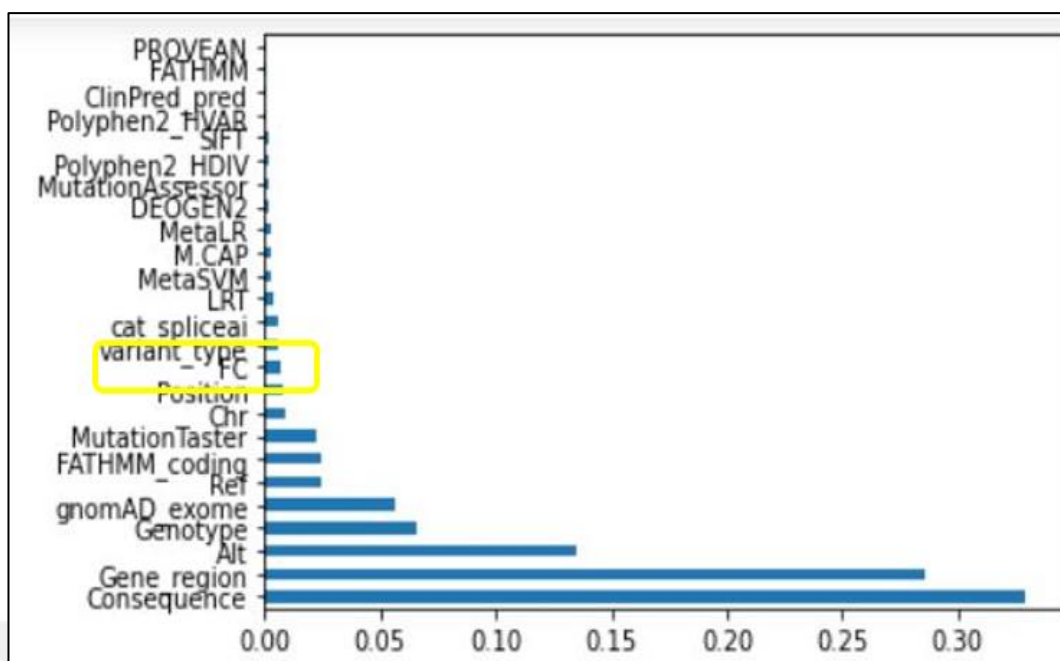


Figure 7.3: Feature importance charts for FC.

The Figure 7.1 and Figure 7.2 was created for categorized FC, and Figure 7.3 was created for numeric FC values. The importance of the feature increases when numeric FC feature is used. Therefore, numerical values were used as the expression feature in the feature table during the model training.

To show the contribution of various features determined within the scope of the thesis, the performance of the model with several set of selected features on balanced and imbalanced data were evaluated. Feature comparison was carried for both models trained with balanced and imbalanced data. The first set group contains chr, position, ref allele, alt allele and variant type defining a variant. This group is called as “main features” since these are most popular features for variant characterization. The feature groups and their corresponding features are shown in Table 7.1.

Table 7.1: Feature group name and features.

Feature Group Name	Features
Main Features	chr, position, ref allele, alt allele and variant type
In silico pred scores	MCAP, SIFT, PolyPhen, LRT, Mutation Tester, Mutation Assessor, MetaSVM, MetaLR, ClinPred, DEOGEN2, PROVEAN, FATHMM, FATHMM_coding,

Table 7.1: Continuation.

VUS_char	Main features + in silico pred scores
VUS_char_expression	Main features + in silico pred scores + FC (expression)
VUS_char_splicing	Main features + in silico pred scores + cat_spliceai (splicing)

According to Figure 7.3, consequence and gene region are the two most effective features in classification. To see the effect of other features, consequence and gene region features were not included in this experiment. Test accuracy of the model on balanced and imbalanced datasets were calculated using 5-fold cross validation presented in the Table 7.2. As a result of the experiment with main features, the model trained with imbalanced data outperformed the model trained with balanced data. With the addition of prediction scores to the main features, the model performance has improved for both models. While the expression and splice features were ineffective in the performance of the model trained with imbalanced data, the expression caused a 1% decrease in the performance of the model trained with balanced data. The splicing feature did not affect the model performance as in the model trained with imbalanced data. According to these scores, the features in the main features group are the most effective on variant characterization, which is supported by the feature importance graph in Figure 7.3.

Table 7.2: Accuracy values of the models trained with various feature groups.

Feature Group Name	path1xbenign10x Test Accuracy	path1xbenign1x Test Accuracy
Main Features	0.97	0.91
VUS_char	0.99	0.97
VUS_char_expression	0.99	0.96
VUS_char_splicing	0.99	0.97

As shown in Figure 7.4, the three most important features in balanced and imbalanced data are Chr, Position and Alt allele in the main feature group. FATHMM_coding and Mutation_Tester from pred scores have a significant effect on the classification performance of both models.

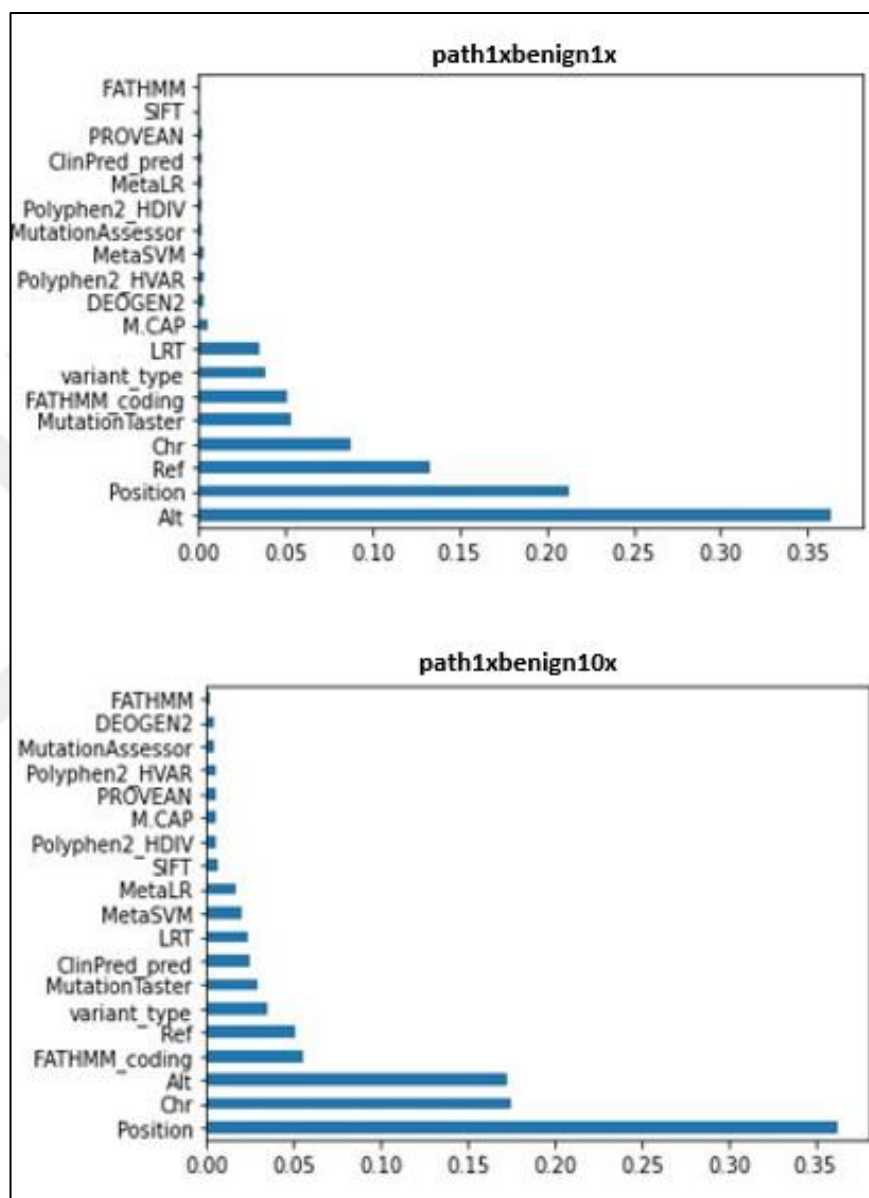


Figure 7.4: Feature importance of the model trained with balanced and imbalanced dataset using features in VUS_char.

According to the literature, variants that increase or decrease gene expression are likely to be pathogenic. So, the effect of variants, referred to as VUS in databases, on gene expressions are manually analysed and the pathogenicity of the variant is usually interpreted accordingly. In the light of these studies, model training was carried

out by adding FC to the features in VUS_char. The feature importance graph obtained as a result of the model trained by features in grouped named VUS_char_expression is shown in Figure 7.5. FC represents expression feature and one of the important features.

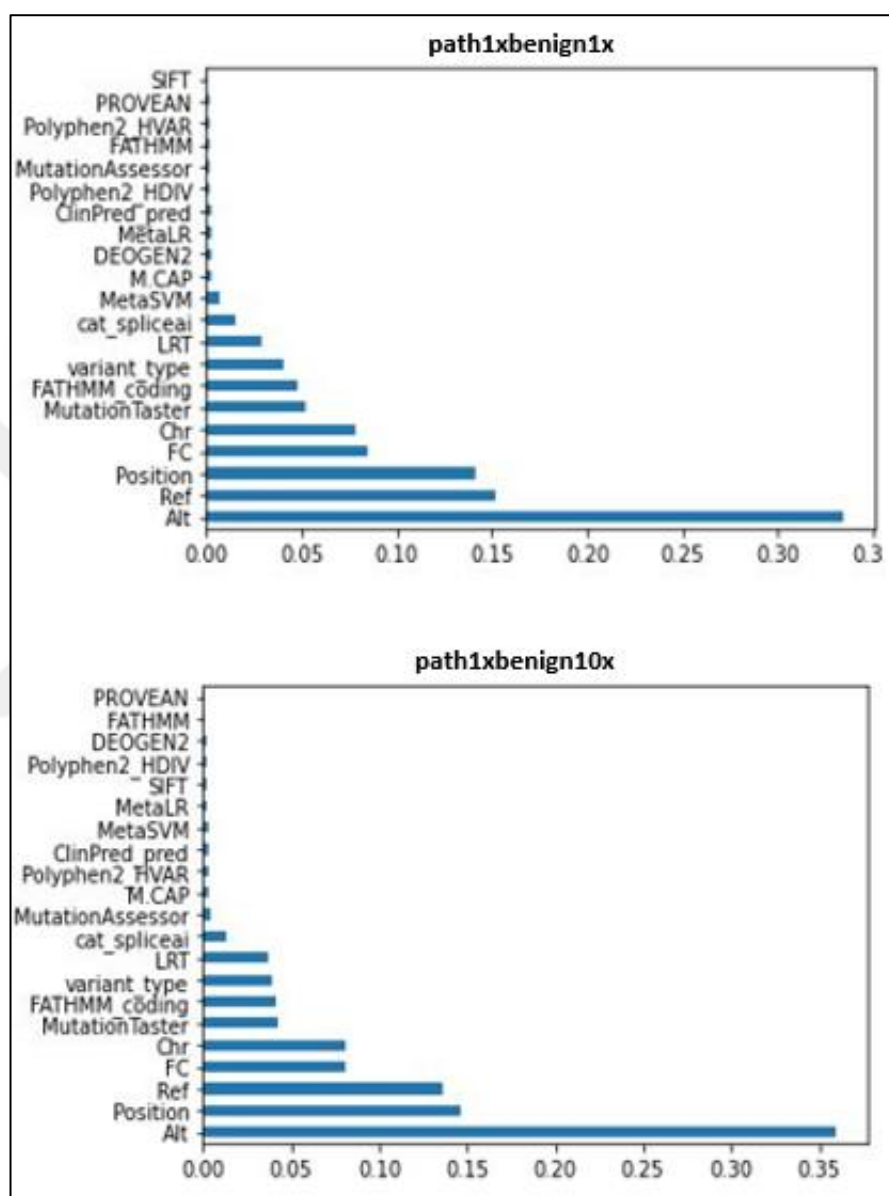


Figure 7.5: Feature importance of the model trained with balanced and imbalanced dataset using features in VUS_char_expression.

According to Table 7.2, when the expression feature was added, a slight decrease was observed in the performance of the model trained with balanced data. However, according to Figure 7.5, expression feature (FC) is one of the features that are effective on variant characterization. So, it is interpreted that although the expression value has

a negative effect on variant characterization, this effect may be due to the FC calculation method or the samples coming from different tissues. The negative effect of its expression on variant characterization was investigated in detail and the results are presented in Section 7.2.

Variants that affect the splicing process and cause aberrant splicing are likely to be pathogenic. In the literature, it has been reported that 60% of the SNPs in the splice site are pathogenic. So, some VUSs can be characterized by manual examination of the effect on splicing, like expression values. Furthermore, it is also among the criteria of ACMG that variants that act on splicing may be pathogenic. Therefore, splicing feature has been added to VUS_char to present the effect of splicing on variant characterization. The feature importance graph obtained as a result of the model trained by features in grouped named VUS_char_splicing is shown in Figure 7.6. The "cat_spliceai" shown in Figure 7.6 represents the splicing feature. As can be seen from the graph, the splicing feature showed similar performance in models trained with balanced and imbalanced datasets.

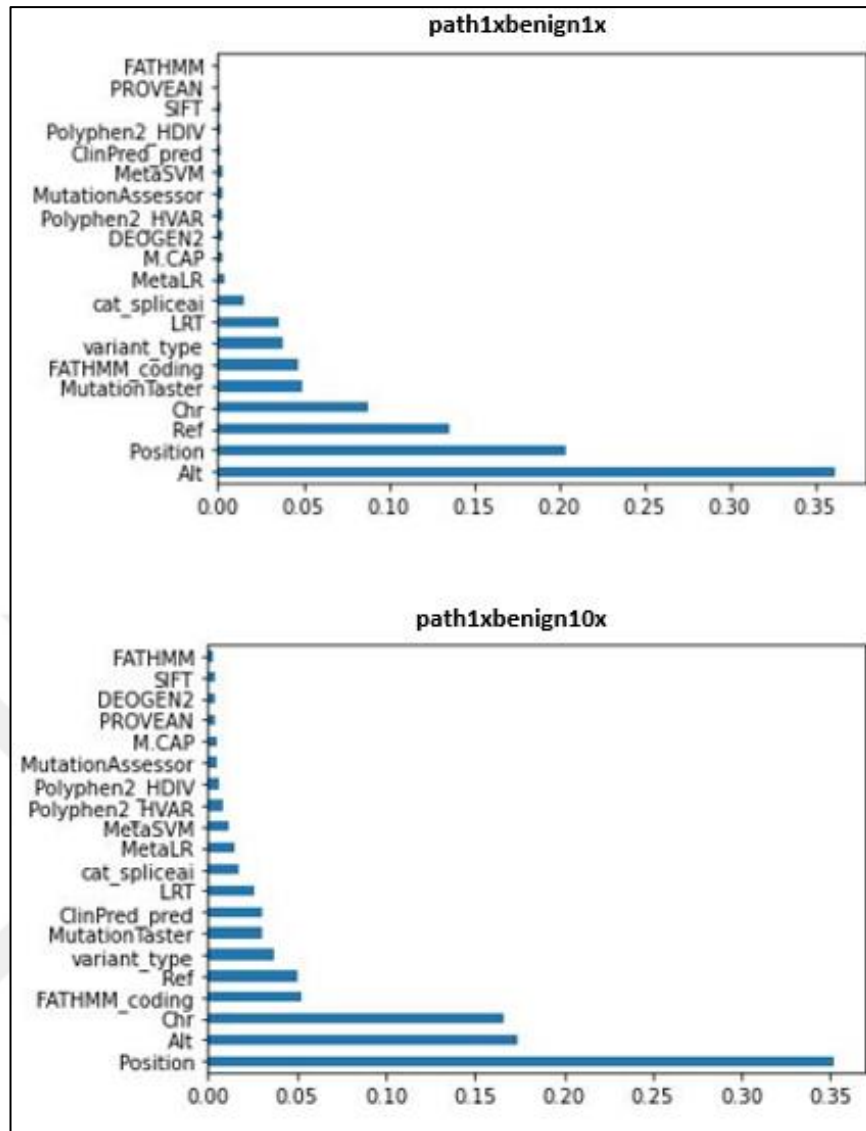


Figure 7.6: Feature importance of the model trained with balanced and imbalanced dataset using features in VUS_char_splicing.

The performance of the models trained with VUS_char_splicing is shown in the Table 7.2. According to this table, there was no significant change in model performance after adding the splicing feature in both models. The reason why splicing has no significant effect on model performance is that the number of variants affecting the splicing process is insufficient for model to learn about them. The results of the study carried out by increasing the number of variants affecting the splicing process can be found in Section 7.3.

7.2. Significance of Expression on Variant Characterization

As mentioned in Section 7.1 direct implication of expression features obtained from the RNA-Seq data is not observed. Additionally, it negatively affects the model performance. So, further experiments were performed to investigate the reasons of this phenomenon.

The FC values were calculated from the gene expressions separately for each dataset and these values were used during model training. For datasets without control data, tissue-specific expression values obtained from Human Protein Atlas were used in FC calculation. This different calculation strategy may have adversely affected the results. To control this effect, either the calculation can be performed using only HPA data or only datasets with control samples can be used during model training.

Among the datasets used within the scope of the thesis, only the dataset GSE136371 does not have tissue specific expression information in HPA. On the other hand, only three datasets have control samples. To have more samples for training, first, tissue specific expressions from HPA were used as controls for the samples in all datasets except the dataset GSE136371. After removing the dataset with ID GSE136371, 2149 pathogenic variants remained in the feature table.

The RNA-Seq data in the datasets used in the study come from different tissues. This may be another reason for the negative effect of the expression feature on model performance. As a solution to this, the tissue types were added as a feature.

Table 7.3: Performance metrics of the model trained with FC calculated using HPA tissue specific expression value.

Feature Group Name	path1xbenign1x Test Accuracy
VUS_char	0.97
VUS_char_expression	0.96
VUS_char_expression + tissue	0.95
Main features	0.91
Main Features + expression	0.90
Main Features + expression + tissue	0.90

As shown in Table 7.3, using HPA tissue-specific expression as a control to calculate FC values does not improve the accuracy and even has a negative effect the model performance. Using the tissue feature besides the expression feature results in worse performance.

In the second option, FC values are calculated only for datasets with control samples. When data were limited to dataset that have control samples, 450 pathogenic variants remained. 450 benign variants were randomly selected to continue model training with a 1:1 ratio.

In previous experiments, the samples were compared with all controls (one vs all), while in this experiment, samples with the same disease were compared with corresponding controls. Raw expression values were normalized using the DESeq tool and FC were recalculated. The same FC value was used for all samples in the datasets. For example, the same FC values for the variants in a gene detected in the samples in the dataset consisting of cystic fibrosis patients are added to the table.

Table 7.4: Performance metrics of the model trained with FC calculated using expression value of controls in datasets.

Feature Group Name	path1xbenign1x Test Accuracy
VUS_char	0.97
VUS_char_expression	0.97
VUS_char_expression + tissue	0.97
Main Features	0.82
Main Features + expression	0.82
Main Features + expression + tissue	0.83

As shown in Table 7.4, after the calculation method were changed, although the model performance was not adversely affected, there is no significant improvement. The reason why the expression feature does not affect model performance may be due to the insufficient number of genes whose expression has changed. Out of 900 variants used for training, 24 genes with 36 variants are down-regulated, and two genes with two variants are up-regulated. Therefore, the number of genes whose expression was changed was insufficient train the model for these variants.

7.3. Significance of Splicing on Variant Characterization

It has been indicated in Section 7.1 that the splicing feature was ineffective on model performance. The reason this result may be the insufficient number of variants that affect the splicing process. In our results, 41 out of 4764 variants in total affect splicing. All the variants' having impact on splicing process are pathogenic. To overcome this problem, a model training was performed using all the variants in the databases.

For this experiment, all the variants found in the InterVar and ClinVar databases were combined. These variants were filtered based on frequency and consequence. A total of 105514 variants remained after the unclassified variants were removed. Splicing analysis was performed for these variants. 1122 variants affecting splicing were identified. Of these variants, 1038 are pathogenic and 84 are benign.

All pathogenic variants affecting splicing are used in the dataset. Pathogenic variants that do not affect splicing was selected as much as pathogenic variants affecting splicing and added into the dataset. As far as the benign variants are concerned, all benign variants affecting splicing are taken. From the remaining benign variants, a random selection is performed until there are same number of benign variants as pathogenic variants. The resulting training set consists of 1996 pathogenic and 1996 benign variants.

Table 7.5: Model performance metrics for splicing feature obtained by splicing analysis of variants in databases.

Feature Group Name	path1xbenign1x Test Accuracy
VUS_char	0.98
VUS_char_splicing	0.98
Main Features	0.63
Main Features + splicing	0.73

The performance comparison of using splicing features in model training besides main features and in silico prediction scores are shown in Table 7.5. The splicing feature did not affect the model performance when it is used in addition to VUS_char

features as in model training with public data. Because variants were classified quite good using their prediction scores, the splice feature was ineffective.

To observe the effect of the splicing feature on the classification, the model is trained by adding the splicing feature to the main features only, without prediction scores. As a result, an increase in the test accuracy of the model was observed. This means that some variants that cannot be characterized correctly with the main features were classified correctly with the splicing feature.

Table 7.6: Comparison of predictions of one of the cross-validation models.

Count	Actual	Pred. Main Features	Pred. Main Features + Splicing	Splicing Effect Yes	Splicing Effect No	Empty
61	Pathogenic	Benign	Pathogenic	54	2	5
37	Pathogenic	Pathogenic	Benign	0	29	8
70	Benign	Pathogenic	Benign	0	53	17
22	Benign	Benign	Pathogenic	10	7	5
81	Benign	Pathogenic	Pathogenic	5	61	15
77	Pathogenic	Benign	Benign	2	55	20

The predictions of one of the models generated during the cross validation is shown in Table 7.6. Accordingly, most of the variants that were incorrectly predicted by the model trained with the main features were predicted correctly when the splicing feature was added. Naturally the model tends to classify variants that influence splicing as pathogenic. The benign variants that the model trained with main features classified as pathogenic were correctly classified by the model with additional the splicing feature.

7.4. Model Comparison and Handling Imbalanced Data

According to graphs of feature importance shown in Figure 7.5, the effects of the features used on the models trained with both balanced and imbalanced datasets are similar. When the performance metrics of the models trained with balanced and imbalanced data were compared (see Table 7.2), two models performed similarly, as well.

However, due to the biased classification of the model, accuracy score may not be a good performance metric in the model trained with imbalanced data. In generally a model trained with imbalanced data classifies the major group in the data better.

In the model trained with imbalanced dataset, the major group is benign. Therefore, the model may tend to classify pathogenic variants as benign. In order to check this, a model was trained using imbalanced data and all features including expression and splicing determined in this thesis. As a result, accuracy, sensitivity scores were 0.99 and specificity scores were 0.98 after 5-fold cross validation. According to these performance metrics, the model is expected to correctly classify all pathogens in the real-world dataset. However, the model results for real-world dataset summarized by the confusion matrix in Figure 7.7 was not good. The sensitivity was 0.98 but none of the pathogenic variants could be classified correctly. The model tends to classify almost all variants as benign. This is an important problem for models trained with imbalanced data.

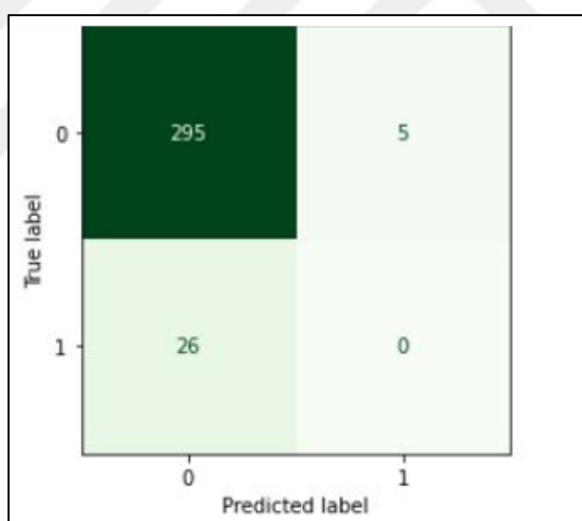


Figure 7.7: Confusion matrix of the model trained with class weight = none.

One problem of the imbalanced data used for the models in the scope of the thesis is that the selected benign variants are quite similar to each other. Therefore, although the number of benign variants has been increased, there was no new information added to the model.

This problem was solved for imbalanced data by increasing the number of min sample splits while the samples are divided into sub-nodes. The minimum number of

sample splits is two in the default settings of Random Forest Classifier. In the experiments, the number of samples split was determined using RandomizedSearchCV from scikit-learn. Many combinations have been tried to determine the number of sample splits manually. RandomizedSearchCV performs random searches on hyperparameters and gives the most appropriate parameter combination as an output. The range of values to be used for the minimum sample split is determined before the use of RandomizedSearchCV.

When using RandomizedSearchCV, all values between 0.1 and 1.0 were tried for the min sample split parameter. As a result, a parameter called min samples split for imbalanced data was determined as 0.5. Model training was carried out with the min sample split parameter set to 0.5 and the other parameters left to be default. As a result, accuracy was 0.91, sensitivity was 0.99, and specificity was 0.08 after cross validation.

The main goal in developing the VUS characterization tool was to accurately classify pathogenic variants. Since pathogenic variants constitute the negative class, in the model performance should be evaluated according to the specificity score. As a result of the model trained by changing the min sample split parameter, the specificity was still found to be very low. This result means that most pathogenic variants cannot be classified correctly. Since sensitivity was high, almost all benign variants are classified correctly. The high sensitivity and very low specificity indicate that the problem of biased classification persists for the model trained with imbalanced data. As a solution to this, the class weight parameter was changed in the model trained with imbalanced data.

Weights scale the loss function. Without weights set, the model treats each sample as equally important. When the class weight is set to 'balanced', the model automatically assigns the class weights inversely proportional to number of members in each class. Class weight can also be adjusted manually. In the model trained with imbalanced data, the performance of correct classification of the minor group by the model is improved by giving a lower weight to the major group and a higher weight to the minor group.

Model training was carried out by using the Random Forest Classifier's min sample split parameter of 0.5, class weight parameter to none, balanced and manual. Other parameters were at their default settings. The performance metrics of these

models after 5-fold cross validation are shown in Table 7.7. “M” represents model in Table 7.7.

Table 7.7: Performance metrics of the model trained by changing class weights.

M	Parameters	Test accuracy	Specificity	Sensitivity
1	class_weight = none	0.91	0.08	0.99
2	class_weight = benign: 1, pathogenic:1	0.91	0.07	0.99
3	class_weight = benign: 1, pathogenic: 2	0.94	0.87	0.95
4	class_weight = balanced	0.94	0.99	0.93
5	class_weight = benign: 1, pathogenic: 5	0.94	0.99	0.94
6	class_weight = benign: 1, pathogenic: 10	0.94	0.99	0.93
7	class_weight = benign: 1, pathogenic: 50	0.90	0.99	0.89
8	class_weight = benign: 1, pathogenic: 100	0.83	0.99	0.81

According to Table 7.7, the models 3, 4, 5 and 6 have the highest and same accuracy. The specificity of the model 3 is slightly lower than the other models with high accuracy. The accuracy and sensitivity of the model 7 and model 8, which have the same specificity as these models, are lower. So, performances of models 4, 5, and 6 are considered to be better than the others.

To select the model with the best performance among these models, models 4, 5 and 6 were tested with real-world dataset. The results of the test with this hold out data are shown in Table 7.8.

Table 7.8: Performance metrics of the models trained by changing class weights for hold-out data.

M	Parameters	accuracy	specificity	sensitivity
4	class_weight = balanced	0.62	0.88	0.60
5	class_weight = benign: 1, pathogenic: 5	0.68	0.42	0.71
6	class_weight = benign: 1, pathogenic: 10	0.64	0.88	0.62

According to Table 7.8, the 5th model presents the highest accuracy. Class weight parameter set to 1 for the benign class and 5 for pathogenic class. Although the specificity of this model was the same as the other two models after cross validation, it was found to be quite low for real-world data compared to the others. The specificities of the model 4 and model 6 are the same and 88%. The sensitivity and

accuracy of the 6th model is slightly higher than the model 4. Since the class weights are determined according to the number of members in class with the balanced parameter, it was expected that the results of these two models would be similar. Among the models trained with imbalanced data, it was determined as the model 6 with the best performance due to a slight difference in accuracy and sensitivity.

Compared to the models trained with imbalanced and balanced data, the model trained with balanced data performed better.

Firstly, balanced data was trained with all the features determined in the thesis and in the default settings of RandomForestClassifier. Later, the training was carried out by setting the min sample split parameter to 0.5 and the other parameters as default. Since the number of members in the classes is balanced, the class weight parameter has not been changed. Table 7.9 shows the performance metrics of the model trained with balanced data after cross validation. “M” represents model. Model 1 has better performance in all three scores, called accuracy, specificity, and sensitivity.

Table 7.9: Performance metrics of the model trained by balanced data.

M	Parameters	Test accuracy	Specificity	Sensitivity
1	with default parameters	0.98	0.98	0.97
2	min samples split=0.5	0.96	0.97	0.94

Both models have been tested with real-world data. The results are shown in Table 7.10. According to these results, model 1 showed the best performance for real-world data, as well.

Table 7.10: Performance metrics of the model trained by balanced data for hold-out data.

M	Parameters	Test accuracy	Specificity	Sensitivity
1	with default parameters	0.70	0.88	0.69
2	min sample split=0.5	0.58	0.96	0.55

When model 6, which performs the best model trained with imbalanced data, and model 1, which performs the best models trained with balanced data, are

compared, model 1 performs slightly better in terms of accuracy and sensitivity, although their specificities are similar. So, model 1, with balanced data was chosen as the model that emerged from the thesis.



8. CONCLUSION AND FUTURE WORK

Clinical characterization of variants is an important problem. Most of the variants obtained by variant analysis are not reported because they are characterized as VUS. Therefore, many VUS variants with the potential to be associated with a disease are overlooked. There are studies in the literature that prioritize VUSs with splicing and expression information obtained from RNA-Seq data. However, using the information obtained in RNA-Seq data for variant characterization requires expertise and manual inspection. So far, there is no variant characterization model based on RNA-Seq data.

In this study, RNA-Seq data was used while developing the variant characterization model. Our model is the first variant classifier model based on machine learning on RNA-Seq data.

RNA-Seq was processed by performing variant calling, annotation of variants, expression and splicing analysis. Outputs obtained from RNA-Seq data analyses constitute the features used in the training set. After the training set was prepared, ML based variant characterization model using random forest classifier was developed. Although the developed model can characterize germline variants with an accuracy of 98%, the major effect of expression and splicing information obtained from RNA-Seq data shown in studies in the literature was not observed in our study. While the expression feature was negatively affected the model performance, adding the splicing feature did not significantly change the model performance.

Based on several experiments performed to analyse the reason for not observing the effect of the expression feature on variant characterization, the most important one is the scarcity in the datasets. Although all datasets in public databases were used in our study, still the number of samples is very limited. Moreover, RNA-Seq data of samples forming each dataset were isolated from different tissues and not every dataset has a control group. So, first, detection of genes with altered expression in downstream analysis was not performed effectively. Secondly, the number of variants from the same tissue is very limited for the model to learn the tissue specific expression changes. Several approaches were tried such as using the tissue feature and calculation of expression changes using tissue specific expression value in databases. However, the expression feature did not significantly affect the model performance.

Another important feature obtained from RNA-Seq data is splicing events. The splicing feature did not significantly change the model performance, neither. The reason is again the low number of variants in the training set that affect the splicing process. In total, 41 of 4764 variants has an effect in the splicing process. To increase the number of variants that have an effect on splicing, a new training set was prepared with variants obtained from databases. On this dataset the importance of the splicing feature on variant characterization is demonstrated. There is around %15 increase in model performance by using splicing featured in addition to main features.

In future, as more RNA-Seq data available, the number of variants that have an effect gene expression values or splicing will be increased. This study has to be repeated to observe improvement in the model performance. Within the scope of the thesis, the expression feature is calculated at the gene level, transcript level expression values can also be used instead to show the effect of this feature on variant characterization. In addition, the selection of benign variants was performed randomly within the scope of the thesis. To increase the contribution various types of benign variants in model training, variants can be clustered according to their attributes selection of adequate variants form each cluster can be assured. Thus, more reasonable distribution and improved variation of benign variants in training set can be achieved. Using this enhanced dataset, the model trials carried out within the scope of the study can be repeated as a future work.

REFERENCES

- Adetunji M. O., Lamont S. J., Abasht B., Schmidt C. J., (2019), “Variant analysis pipeline for accurate detection of genomic variants from transcriptome sequencing data”, *PLOS ONE*, 14(9).
- Adzhubei I., Schmidt S., Peshkin L., Ramensky V. E., Gerasimova A., Bork P., Kondrashov A. S., Sunyaev S. R., (2010), “A method and server for predicting damaging missense mutations”, *Nat Methods*, 7, 248–249.
- Alirezaie N., Kernohan K. D., Hartley T., Majewski J., Hocking T.D., (2018), “ClinPred: Prediction Tool to Identify Disease-Relevant Nonsynonymous Single-Nucleotide Variants”, *Am J Hum Genet*, 103(4), 474–483.
- Anders S., Pyl P.T., Huber W., (2015), “HTSeq--a Python framework to work with high-throughput sequencing data”, *Bioinformatics*, 31(2), 166–169.
- Anna A., Monika G., (2018), “Splicing mutations in human genetic disorders: examples, detection, and confirmation”, *Journal of Applied Genetics*, 59(3), 253–268.
- Auwerda G. A., Carneiro M. O., Hartl C., Poplin R., del Angel G., Levy-Moonshine A., Jordan T., Shakir K., Roazen D., Thibault J., Banks E., Garimella K. V., Altshuler D., Gabriel S., DePristo M. A., (2013a), “From FastQ Data to High-Confidence Variant Calls: The Genome Analysis Toolkit Best Practices Pipeline”, *Current Protocols in Bioinformatics*, 43(1).
- Auwerda G. A., Carneiro M. O., Hartl C., Poplin R., del Angel G., Levy-Moonshine A., Jordan T., Shakir K., Roazen D., Thibault J., Banks E., Garimella K. V., Altshuler D., Gabriel S., DePristo M. A., (2013b), “From FastQ Data to High-Confidence Variant Calls: The Genome Analysis Toolkit Best Practices Pipeline”, *Current Protocols in Bioinformatics*, 43(1).
- Baruzzo G., Hayer K. E., Kim E. J., di Camillo B., FitzGerald G. A., Grant G. R., (2017), “Simulation-based comprehensive benchmarking of RNA-seq aligners”, *Nature Methods*, 14(2), 135–139.
- Baten A., Chang B., Halgamuge S. Li J., (2006), “Splice site identification using probabilistic parameters and SVM classification”, *BMC Bioinformatics*, 7(S5), S15.
- Bolger A. M., Lohse M., Usadel B., (2014), “Trimmomatic: a flexible trimmer for Illumina sequence data”, *Bioinformatics*, 30(15), 2114–2120.
- Bray N. L., Pimentel H., Melsted P., Pachter L., (2016), “Near-optimal probabilistic RNA-seq quantification”, *Nature Biotechnology*, 34(5), 525–527.
- Breiman L., (2001), “Random Forests”, *Machine Learning*, 45(1), 5–32.
- Choi H.Y., Jo H., Zhao X., ... Hayes D. N., (2021), “SCISSOR: a framework for identifying structural changes in RNA transcripts”, *Nat Commun*, 12, 286.

Choi Y., Sims G. E., Murphy S., Miller J. R., Chan A. P., (2012), “Predicting the functional effect of amino acid substitutions and indels”, *PLoS One*, 7(10).

Chun S., Fay J.C., (2009), “Identification of deleterious mutations within three human genomes”, *Genome Res*, 19(9), 1553-61.

Cingolani P., Platts A., Wang L. L., Coon M., Nguyen T., Wang L., Land S. J., Lu X., Ruden D. M., (2012), “A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff”, *Fly*, 6(2), 80–92.

Clancy S., (2008), “RNA splicing: introns, exons and spliceosome”, *Nature Education*, 1.

Conesa A., Madrigal P., Tarazona S., Gomez-Cabrero D., Cervera A., McPherson A., Szczesniak M. W., Gaffney D. J., Elo L. L., Zhang X., Mortazavi A., (2016), “A survey of best practices for RNA-seq data analysis”, *Genome Biology*, 17(1), 13.

CRICK F., (1970), “Central Dogma of Molecular Biology”, *Nature* 227, 561–563.

Cummings B.B., Marshall J.L., Tukiainen T., ... MacArthur D.G., (2017), “Improving genetic diagnosis in Mendelian disease with transcriptome sequencing”, *Sci Transl Med*, 9(386).

Curry P. D. K., Broda K. L., Carroll C. J., (2021), “The Role of RNA-Sequencing as a New Genetic Diagnosis Tool”, *Current Genetic Medicine Reports*, 9(2), 13–21.

Dahariya S., Paddibhatla I., Kumar S., Raghuwanshi S., Palapati A., Gutti R. K., (2019), “Long non-coding RNA: Classification, biogenesis and functions in blood cells”, *Mol Immunol*, 112, 82-92.

Dai M., Thompson R. C., Maher C., Contreras-Galindo R., Kaplan M. H., Markovitz D. M., Omenn G., Meng F., (2010), “NGSQC: cross-platform quality analysis pipeline for deep sequencing data”, *BMC Genomics*, 11(S4), S7.

Davies J. C., Alton E. W. F. W., Bush A., (2007), “Cystic fibrosis”, *BMJ*, 335(7632), 1255–1259.

Dobin A., Davis C. A., Schlesinger F., Drenkow J., Zaleski C., Jha S., Batut P., Chaisson M., Gingeras T. R., (2013), “STAR: ultrafast universal RNA-seq aligner”, *Bioinformatics*, 29(1), 15–21.

Dong C., Wei P., Jian X., Gibbs R., Boerwinkle E., Wang K., Liu X., (2015), “Comparison and integration of deleteriousness prediction methods for nonsynonymous SNVs in whole exome sequencing studies”, *Hum Mol Genet*, 24(8), 2125-37.

Evans C., Hardin J., Stoebe D. M., (2018), “Selecting between-sample RNA-Seq normalization methods from the perspective of their assumptions”, *Briefings in Bioinformatics*, 19(5), 776–792.

Frankish A., Diekhans M., Ferreira A.-M., Johnson R., Jungreis I., Loveland J., Mudge J. M., Sisu C., Wright J., Armstrong J., Barnes I., Berry A., Bignell A., Sala S. C.,

Chrast J., Cunningham F., Di Domenico T., Donaldson S., Fiddes I. T., ... Flicek P., (2019), "GENCODE reference annotation for the human and mouse genomes", *Nucleic Acids Research*, 47, 766–773.

Edwards G., (2018), "Machine Learning | An Introduction", Towards Data Science, <https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0>.

Gallego R., I., Pai A. A., Tung J., Gilad Y., (2014), "RNA-seq: impact of RNA degradation on transcript quantification", *BMC Biol*, 12.

Ghosh R., Harrison S. M., Rehm H. L., Plon S. E., Biesecker L. G.; ClinGen Sequence Variant Interpretation Working Group., (2018), "Updated recommendation for the benign stand-alone ACMG/AMP criterion", *Hum Mutat*, 39(11), 1525-1530.

Goswami C., Chattopadhyay A., Chuang E. Y., (2021), "Rare variants: data types and analysis strategies", *Ann Transl Med*, 9(12), 961.

Haddad C. F., (2020), "Hereditary breast cancer: review and current approach", *Mastology*, 30.

Henrie A, Hemphill S.E, Ruiz-Schultz N., Cushman B., DiStefano M.T., Azzariti D., Harrison S.M., Rehm H.L., Eilbeck K., (2018), "ClinVar Miner: Demonstrating utility of a Web-based tool for viewing and filtering ClinVar data", *Hum Mutat.*, 39(8), 1051-1060.

Hershy A., (2019), "Gini Index vs Information Entropy", Towards Data Science, <https://towardsdatascience.com/gini-index-vs-information-entropy-7a7e4fed3fcb>.

Jaganathan K., Kyriazopoulou Panagiotopoulou S., McRae J. F., Darbandi S. F., Knowles D., Li Y. I., Kosmicki J. A., Arbelaez, J., Cui W., Schwartz G. B., Chow E. D., Kanterakis E., Gao H., Kia A., Batzoglou S., Sanders S. J., Farh K. K.-H., (2019), "Predicting Splicing from Primary Sequence with Deep Learning", *Cell*, 176(3), 535-548.e24.

Karki R., Pandya D., Elston R. C., Ferlini C., (2015), "Defining "mutation" and "polymorphism" in the era of personal genomics", *BMC Medical Genomics*, 8(1), 37.

Keel B. N., Snelling W. M., (2018), "Comparison of Burrows-Wheeler Transform-Based Mapping Algorithms Used in High-Throughput Whole-Genome Sequencing: Application to Illumina Data for Livestock Genomes", *Frontiers in Genetics*, 9.

Kim D., Langmead B., Salzberg S. L., (2015), "HISAT: a fast spliced aligner with low memory requirements", *Nature Methods*, 12(4), 357–360.

Kim P., Jang Y.E., Lee S., (2019), "FusionScan: accurate prediction of fusion genes from RNA-Seq data", *Genomics Inform*, 17(3), 26.

Koboldt D. C., Chen K., Wylie T., Larson D. E., McLellan M. D., Mardis E. R., Weinstock G. M., Wilson R. K., Ding L., (2009), "VarScan: variant detection in massively parallel sequencing of individual and pooled samples", *Bioinformatics*, 25(17), 2283–2285.

Kremer L., Bader D., Mertes C., et al., (2017), “Genetic diagnosis of Mendelian disorders via RNA sequencing”, *Nat Commun*, 8, 15824.

Lai C., Zimmer A. D., O’Connor R., Kim S., Chan R., Akker J., Zhou A. Y., Topper S., Mishne G., (2020), “LEAP: Using machine learning to support variant classification in a clinical setting”, *Human Mutation*, 41(6), 1079–1090.

Landrum M.J., Lee J.M., Riley G.R., Jang W., Rubinstein W.S., Church D.M., Maglott D.R., (2014), “ClinVar: public archive of relationships among sequence variation and human phenotype”, *Nucleic Acids Res*, 42, 980-985.

Langmead B., Trapnell C., Pop M., Salzberg S. L., (2009), “Ultrafast and memory-efficient alignment of short DNA sequences to the human genome”, *Genome Biology*, 10(3), R25.

Li H., Durbin R., (2009), “Fast and accurate short read alignment with Burrows-Wheeler transform”, *Bioinformatics*, 25(14), 1754–1760.

Li H., Handsaker B., Wysoker A., Fennell T., Ruan J., Homer N., Marth G., Abecasis G., Durbin R., (2009a), “The Sequence Alignment/Map format and SAMtools”, *Bioinformatics*, 25(16), 2078–2079.

Li H., Handsaker B., Wysoker A., Fennell T., Ruan J., Homer N., Marth G., Abecasis G., Durbin R., (2009b), “The Sequence Alignment/Map format and SAMtools”, *Bioinformatics*, 25(16), 2078–2079.

Li Q., Wang K., (2017), “InterVar: Clinical Interpretation of Genetic Variants by the 2015 ACMG-AMP Guidelines”, *The American Journal of Human Genetics*, 100(2), 267–280.

Liao Y., Smyth G. K., Shi W., (2014), “featureCounts: an efficient general purpose program for assigning sequence reads to genomic features”, *Bioinformatics*, 30(7), 923–930.

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), 550.

Lynch H., Lynch P., Lanspa S., Snyder C., Lynch J., Boland C., (2009), “Review of the Lynch syndrome: history, molecular genetics, screening, differential diagnosis, and medicolegal ramifications”, *Clinical Genetics*, 76(1), 1–18.

Marco-Puche G., Lois S., Benítez J., Trivino J. C., (2019), “RNA-Seq Perspectives to Improve Clinical Diagnosis”, *Frontiers in Genetics*, 10.

Martin M. (2011), “Cutadapt removes adapter sequences from high-throughput sequencing reads”, *EMBnet.Journal*, 17(1), 10.

McKenna A., Hanna M., Banks E., Sivachenko A., Cibulskis K., Kernytsky A., Garimella K., Altshuler D., Gabriel S., Daly M., DePristo M. A., (2010), “The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data”, *Genome Research*, 20(9), 1297–1303.

McLaren W., Gil L., Hunt S. E., Riat H. S., Ritchie G. R. S., Thormann A., Flicek P., Cunningham F., (2016), “The Ensembl Variant Effect Predictor”, *Genome Biology*, 17(1), 122.

Meola G., Cardani R., (2015), “Myotonic dystrophies: An update on clinical aspects, genetic, pathology, and molecular pathomechanisms”, *Biochimica et Biophysica Acta (BBA) - Molecular Basis of Disease*, 1852(4), 594–606.

Mertes C., Scheller I. F., Yépez V. A. et al., (2021), “Detection of aberrant splicing events in RNA-seq data using FRASER”, *Nat Commun*, 12, 529.

Mottaghinejad S., (2021), “The intuition behind bias and variance”, *Towards Data Science*. <https://towardsdatascience.com/bias-and-variance-but-what-are-they-really-ac539817e171>.

Murdock D.R., Dai H., Burrage L.C., Rosenfeld J.A., Ketkar S., Müller M.F., Yépez V.A., Gagneur J., Liu P., Chen S., Jain M., Zapata G., Bacino C.A., Chao H.T., Moretti P., Craig W.J., Hanchard N.A.; Undiagnosed Diseases Network, Lee B., (2021), “Transcriptome-directed analysis for Mendelian disease diagnosis overcomes limitations of conventional genomic testing”, *J Clin Invest*, 131(1), 141500.

Musich R., Cadle-Davidson L., Osier M. V., (2021), “Comparison of Short-Read Sequence Aligners Indicates Strengths and Weaknesses for Biologists to Consider”, *Frontiers in Plant Science*, 12.

Needleman S. B., Wunsch C. D., (1970), “A general method applicable to the search for similarities in the amino acid sequence of two proteins” *Journal of Molecular Biology*, 48(3), 443–453.

Neums L., Suenaga S., Beyerlein P., Anders S., Koestler D., Mariani A., Chien J., (2018), “VaDiR: an integrated approach to Variant Detection in RNA”, *GigaScience*, 7(2).

Ng P. C., Henikoff S., (2003), “SIFT: predicting amino acid changes that affect protein function”, *Nucleic Acids Research*, 31(13), 3812–3814.

Nicora G., Limongelli I., Gambelli P., Memmi M., Malovini A., Mazzanti A., Napolitano C., Priori S., Bellazzi R., (2018), “CardioVAI: An automatic implementation of ACMG-AMP variant interpretation guidelines in the diagnosis of cardiovascular diseases”, *Human Mutation*, 39(12), 1835–1846.

Patel R. K., Jain M., (2012), “NGS QC Toolkit: A Toolkit for Quality Control of Next Generation Sequencing Data”, *PLoS ONE*, 7(2), e30619.

Pertea M., Pertea G. M., Antonescu C. M., Chang T.-C., Mendell J. T., Salzberg S. L., (2015), “StringTie enables improved reconstruction of a transcriptome from RNA-seq reads”, *Nature Biotechnology*, 33(3), 290–295.

Qi Y., (2012), “Random Forest for Bioinformatics”, *Ensemble Machine Learning*, 307–323.

Raimondi D., Tanyalcin I., Ferté J., Gazzo A., Orlando G., Lenaerts T., Rooman M., Vranken W., (2017), “DEOGEN2: prediction and interactive visualization of single amino acid variant deleteriousness in human proteins”, *Nucleic Acids Res*, 45, 201-206.

Ramensky V., (2002), “Human non-synonymous SNPs: server and survey”, *Nucleic Acids Research*, 30(17), 3894–3900.

Ravichandran V., Shameer Z., Kemel Y., Walsh M., Cadoo K., Lipkin S., Mandelker D., Zhang L., Stadler Z., Robson M., Offit K., Vijai J., (2019), “Toward automation of germline variant curation in clinical cancer genetics”, *Genetics in Medicine*, 21(9), 2116–2125.

Rawat T., Khemchandani V., (2017), “Feature Engineering (FE) Tools and Techniques for Better Classification Performance”, *International Journal of Innovations in Engineering and Technology*, 8(2).

Rentzsch P., Schubach M., Shendure J., Kircher M., (2021), “CADD-Splice--improving genome-wide variant effect prediction using deep learning-derived splice scores”, *Genome Med*, 13.

Reva B., Antipin Y., Sander C., (2011), “Predicting the functional impact of protein mutations: application to cancer genomics”, *Nucleic Acids Res*, 39(17), 118.

Richards S., Aziz N., Bale S., Bick D., Das S., Gastier-Foster J., Grody W. W., Hegde M., Lyon E., Spector E., Voelkerding K., Rehms H. L., (2015), “Standards and guidelines for the interpretation of sequence variants: a joint consensus recommendation of the American College of Medical Genetics and Genomics and the Association for Molecular Pathology”, *Genetics in Medicine*, 17(5), 405–424.

Ritchie M. E., Phipson B., Wu D., Hu Y., Law C. W., Shi W., Smyth G. K., (2015), “limma powers differential expression analyses for RNA-sequencing and microarray studies”, *Nucleic Acids Research*, 43(7), e47–e47.

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.

Sanger F., Nicklen S., Coulson A.R., (1977), “DNA sequencing with chain-terminating inhibitors”, *Proc Natl Acad Sci U S A*, 74(12), 5463-7.

Schroeder A., Mueller O., Stocker S., et al., (2006), “The RIN: an RNA integrity number for assigning integrity values to RNA measurements”, *BMC Molecular Biol* 7(3).

Schwarz J.M., Rödelberger C., Schuelke M., Seelow D., (2010), “MutationTaster evaluates disease-causing potential of sequence alterations”, *Nat Methods*, 7(8), 575-6.

Shihab H. A., Gough J., Cooper D. N., Stenson P. D., Barker G. L., Edwards K. J., Day I. N., Gaunt T. R., (2013), “Predicting the functional, molecular, and phenotypic

consequences of amino acid substitutions using hidden Markov models”, *Hum Mutat*, 34(1), 57-65.

Shihab H. A., Rogers M. F., Gough J., Mort M., Cooper D. N., Day I. N., Gaunt T. R., Campbell C., (2015), “An integrative approach to predicting the functional effects of non-coding and coding sequence variation”, *Bioinformatics*, 31(10), 1536-43.

Slatko B. E., Gardner A. F., Ausubel F. M., (2018), “Overview of Next-Generation Sequencing Technologies”, *Current Protocols in Molecular Biology*, 122(1).

Smith T. F., Waterman M. S., (1981), “Identification of common molecular subsequences”, *Journal of Molecular Biology*, 147(1), 195–197.

Stupnikov A., McInerney C. E., Savage K. I., McIntosh S. A., Emmert-Streib F., Kennedy R., Salto-Tellez M., Prise K. M., McArt D. G., (2021), “Robustness of differential gene expression analysis of RNA-seq”, *Computational and Structural Biotechnology Journal*, 19, 3470–3481.

The 1000 Genomes Project Consortium, (2010), “A map of human genome variation from population-scale sequencing”, *Nature*, 467(7319), 1061–1073.

Trivedi U. H., Căzard T., Bridgett S., Montazam A., Nichols J., Blaxter M., Gharbi K., (2014), “Quality control of next-generation sequencing data without a reference”, *Frontiers in Genetics*, 5.

“The Variant Call Format (VCF) Version 4.1 Specification”, (2021, july 27). <https://samtools.github.io/hts-specs/VCFv4.1.pdf>.

Wai H. A., Lord J., Lyon M., Gunning A., Kelly H., Cibir P., Seaby E. G., Spiers-Fitzgerald K., Lye J., Ellard S., Thomas N. S., Bunyan D. J., Douglas A. G. L., Baralle D., (2020), “Blood RNA analysis can increase clinical diagnostic rate and resolve variants of uncertain significance”, *Genetics in Medicine*, 22(6), 1005–1014.

Wang K., Li M., Hakonarson H., (2010), “ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data”, *Nucleic Acids Research*, 38(16), e164–e164.

Wang Y, Liu J, Huang B.O, Xu Y.M, Li J, Huang L.F, Lin J, Zhang J, Min Q.H, Yang W.M, Wang X.Z., (2015), “Mechanism of alternative splicing and its regulation”, *Biomedical Reports*, 3(2), 152–158.

Yépez V. A., Gusic M., Kopajtich R., Mertes C., Smith N. H., Alston C. L., Ban R., Beblo S., Berutti R., Blessing H., Ciara E., Distelmaier F., Freisinger P., Häberle J., Hayflick S. J., Hempel M., Itkis Y. S., Kishita Y., Klopstock T., ... Prokisch H., (2022), “Clinical implementation of RNA sequencing for Mendelian disease diagnostics”, *Genome Medicine*, 14(1), 38.

Zeng Z., Bromberg Y., (2019), “Predicting Functional Effects of Synonymous Variants: A Systematic Review and Perspectives”, *Frontiers in Genetics*, 10.

Zhao S., Ye Z., Stanton R., (2020), “Misuse of RPKM or TPM normalization when comparing across samples and sequencing protocols”. *RNA*, 26(8), 903–909.

Zhao Y., Li M.-C., Konaté M. M., Chen L., Das B., Karlovich C., Williams P. M., Evrard Y. A., Doroshow J. H., McShane L. M., (2021), “TPM, FPKM, or Normalized Counts? A Comparative Study of Quantification Measures for the Analysis of RNA-seq Data from the NCI Patient-Derived Models Repository”, *Journal of Translational Medicine*, 19(1), 269.

Zheng J.T., Lin C.X., Fang Z.Y., Li H.D., (2020), “Intron Retention as a Mode for RNA-Seq Data Analysis”, *Front Genet.*,11,586.

Xiong H. Y., Alipanahi B., Lee L. J., Bretschneider H., Merico D., Yuen R. K., Hua Y., Gueroussov S., Najafabadi H. S., Hughes T. R., Morris Q., Barash Y., Krainer A. R., Jovic N., Scherer S. W., Blencowe B. J., Frey B. J., (2015), “RNA splicing. The human splicing code reveals new insights into the genetic determinants of disease”, *Science*, 347(6218), 1254806.



BIOGRAPHY

Özgenur AKDAŞ has graduated from Yıldız Technical University Faculty of Arts and Sciences in Molecular Biology and Genetics in the year 2018. She is an M.Sc. student in Bioinformatics and Systems Biology Programme at Gebze Technical University Graduate School of Natural and Applied Sciences.

