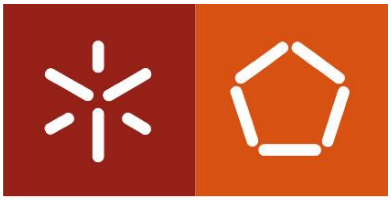


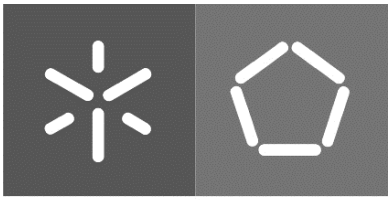
Universidade do Minho

Escola de Engenharia

Ana Rita Patrício Monteiro

Advanced Text Mining for Annotation of Genomic Variants

October 2018



Universidade do Minho

Escola de Engenharia

Ana Rita Patrício Monteiro

Advanced Text Mining for Annotation of Genomic Variants

M.Sc. Thesis on Bioinformatics

Supervisor: Drª Patrícia Oliveira, I3S, University of Porto

Co-Supervisor: Dr. Rui Mendes, University of Minho

October 2018

Anexo 3

DECLARAÇÃO

Nome

Ana Rita Patrício Monteiro

Endereço electrónico: anaritapatriciomonteiro@gmail.com Telefone: 968860036 / _____

Número do Bilhete de Identidade: 14674462

Título dissertação ☒/tese ☐

Advanced Text Mining for Annotation of Genomic variantes

Orientador(es):

Patrícia Joana Morais Ferreira Oliveira; Rui Manuel Ribeiro Castro Mendes Ano de conclusão: 2018

Designação do Mestrado ou do Ramo de Conhecimento do Doutoramento:

Mestrado em Bioinformática - Ramo em Tecnologias da Informação

Nos exemplares das teses de doutoramento ou de mestrado ou de outros trabalhos entregues para prestação de provas públicas nas universidades ou outros estabelecimentos de ensino, e dos quais é obrigatoriamente enviado um exemplar para depósito legal na Biblioteca Nacional e, pelo menos outro para a biblioteca da universidade respectiva, deve constar uma das seguintes declarações:

1. É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE/TRABALHO APENAS PARA EFEITOS D INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL S COMPROMETE;

2. É AUTORIZADA A REPRODUÇÃO PARCIAL DESTA TESE/TRABALHO (indicar, caso tal seja necessário, nº máximo de páginas, ilustrações, gráficos, etc.), APENAS PARA EFEITOS DE INVESTIGAÇÃO, , MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

3. DE ACORDO COM A LEGISLAÇÃO EM VIGOR, NÃO É PERMITIDA A REPRODUÇÃO DE QUALQUER PARTE DESTA TESE/TRABALHO

Universidade do Minho, 26 / 10 / 2018

Assinatura: Ana Rita Patrício Monteiro

Agradecimentos

Uma tese de mestrado é uma longa caminhada, que apesar de às vezes solitária não poderia ter sido trilhada sem o apoio, energia e força de várias pessoas, a quem estou imensamente grata.

À Doutora Carla Oliveira, por me ter dado a oportunidade de desenvolver este projeto de Tese no seu grupo de investigação. Obrigada por toda a simpatia e amabilidade a que me foi habituando e por todas as ideias e conhecimento que me transmitiu e que sem a qual não teria sido capaz de adquirir.

Um especial agradecimento à minha orientadora Doutora Patrícia Oliveira que me guiou nesta caminhada que nem sempre foi fácil de percorrer. Um obrigada é pouco, por todo o esforço e conhecimento que me transmitiu, sempre com uma visão critica e oportuna, o que permitiu desenvolver e enriquecer todas as etapas subjacentes a este projeto. Agradeço a alegria e o entusiasmo contagiante que sempre demonstrou pelos meus resultados e descobertas, o que me levou a querer descobrir e aprender mais.

Ao Doutor Rui Mendes pelo disponibilidade e coorientação demonstrada neste projeto. À Diana Lemos, por toda a ajuda, colaboração e participação neste projeto. Sem o teu apoio, críticas e sugestões este projeto não teria sido possível.
Ao Doutor Pedro Ferreira, por todos os ensinamentos dados, principalmente numa área que tanto me fascina, *Machine Learning*.
Um obrigada enorme também a todos os outros elementos do grupo ERiC por toda a simpatia e companheirismo a que sempre me habituaram.

Por fim, e tendo consciência que sozinha nada disto teria sido possível, dirijo um agradecimento aos meus pais, avó e namorado. Obrigada por serem modelos de coragem e por todo o incentivo, amizade e paciência demonstrados ao longo desta caminhada. Não teria conseguido superar todos os obstáculos que surgiram ao longo desta caminhada sem o vosso incondicional apoio.

Resumo

Título: Text Mining Avançado para Anotação de Variantes Genómicas

A deteção de variantes genómicas associadas à doença tornou-se uma tarefa acessível por meio do sequenciamento de nova geração. Esta tecnologia produz grandes quantidades de dados que usando ferramentas de bioinformática permite entender o impacto funcional das variantes. Contudo, às vezes essas informações estão ocultas em textos clínicos não estruturados, sem uma classificação do tipo 'Benigna' ou 'Patogénica'. Embora tais textos estejam na OMIM, as variantes são frequentemente descritas como 'Variantes de Significado Desconhecido' (VUS). Portanto, para interpretar as informações destes textos desenvolvemos uma ferramenta baseada em *Text-Mining* (TM)/*Machine Learning* (ML). E, recolhemos textos clínicos não estruturados com uma classificação da ClinVar de 'Benignas' ou 'Patogénicas'. E construímos um conjunto de dados com 24.171 textos clínicos não estruturados, onde 174 são de variantes 'Benignas' e 23.997 de variantes 'Patogénicas'. Os textos de cada variante, foram pré-processados para remover informações irrelevantes. Em seguida, construímos um dicionário de palavras-chave biológicas, dando um valor positivo às palavras-chave com uma conotação positiva e um valor negativo às palavras-chave com uma conotação negativa. Assim, aperfeiçoámos uma estratégia única de pontuação para uma máxima *accuracy* na classificação. Para testar a nossa estratégia de pontuação, usámos os textos de todas as variantes 'Benignas' (*n*=174) e 1000 variantes 'Patogénicas' selecionadas aleatoriamente. A análise feita pela nossa ferramenta a 235 textos levou a uma *accuracy* de 89,4%. Finalmente, e usando um conjunto de dados de validação com 10 'Benignas' e 690 'Patogénicas' (*n*=700), conseguimos obter uma *accuracy* de 99%, ou seja, apenas 7 variantes incorretamente classificadas. Em conclusão, a nossa ferramenta é capaz de interpretar e classificar textos da OMIM com uma alta *accuracy*. No futuro, abordaremos as variantes VUS/não classificadas, com o objetivo de fornecer ao utilizador uma probabilidade de que tais variantes sejam 'Benignas' ou 'Patogénicas' num dado contexto de doença.

Palavras-Chave: Variantes Genómicas, *Text Mining*, *Machine Learning*, Classificação de Variantes

Abstract

Title: Advanced Text Mining for Annotation of Genomic Variants

The detection of genomic variants associated with disease has become an accessible task through Next Generation Sequencing. This technology produces large amounts of data that, using bioinformatics tools, allow to understand the functional impact of detected variants. However, in sometimes such information is concealed within unstructured texts (UT) rather than in a binary classification, *i.e.* 'Benign' *vs.* 'Pathogenic'. Although UTs are available in OMIM, in many cases, the variants are described as 'Variants of Unknown Significance' (VUS). Therefore, to interpret the information from UTs, we have designed a *Text-Mining* (TM)/*Machine Learning* (ML)-based tool. To create our tool, we collected OMIM-UTs from a set of ClinVar-classified 'Benign' and 'Pathogenic' genomic variants, constructing a dataset of 24,171 variants, 174 classified by ClinVar as 'Benign' and 23,997 as 'Pathogenic' and the corresponding OMIM-UTs were first pre-processed to remove irrelevant non-clinical information. Next, we constructed a dictionary of biological keywords, giving a positive value to keywords with a positive connotation and a negative value to keywords with a negative connotation a negative or positive connotation to be searched in the OMIM-UTs. Therefore, we fine-tuned a unique scoring strategy for maximum variant-classification accuracy. To train and test we used the corresponding OMIM-UTs of all 'Benign' variants (*n*=174) and 1000 randomly selected 'Pathogenic' variants from our dataset. Classification of OMIM-UTs from the ML-test dataset (*n*=235) by our tool, led to an 89.4% accuracy rate. Finally, and using a validation dataset with 10 'Benign' and 690 'Pathogenic' (*n*=700) we were able to obtain an accuracy rate of 99%, *i.e.* only 7 misclassified variants. In conclusion, our tool is currently capable of classifying OMIM-UTs with a high accuracy rate. In the future, we expect to address the problem of VUS/unclassified variants, aimed at providing the user with a likelihood of whether such variants are more probable to be 'Benign' or 'Pathogenic' in a given disease context.

Keywords: Genomic Variants, *Text Mining*, *Machine Learning*, Variants Classifications

Index

List of Figures

List of Tables

List of Acronyms

ACMG	American College of Medical Genetics and Genomics
AIDS	Acquired Immune Deficiency Syndrome
bp	base pair
CD	Coding Sequences
CNV	Copy Number Variant
COSMIC	Catalogue Of Somatic Mutations In Cancer
DARNED	DAtabase of RNa EDiting in humans
dbGaP	Database of Genotypes and Phenotypes
dbSNP	Single Nucleotide Polymorphism Database
dbVar	Database of Genomic Structural Variation
DDBJ	DNA Data Bank of Japan
DGVa	Database of Genomic Variants archive
DNA	Deoxyribonucleic Acid
DT	Data Mining
EBI	European Bioinformatics Institute
EMBL	European Molecular Biology Laboratory
eQTL	expression Quantitative Trait Loci
ExAC	Exome Aggregation Consortium
FN	False-Negatives
FP	False-Positives
GC	Guanine-cytosine
GWAS	Genome-Wide Association Studies

HGP	Human Genome Project
HIV	Human Immunodeficiency Virus
HPO	Human Phenotype Ontology
IDF	Inverse Document Frequency
IE	Information Extraction
INDELS	Insertion/Deletion of nucleotides
IR	Information Retrieval
LoF	Loss of function
Mb	Megabases
MeSH	Medical Subject Headings
ML	*Machine Learning*
MNV	Multi-Nucleotide Changes
NCBI	National Centre for Biotechnology Information
NER	Named Entity Recognition
NGS	Next Generation Sequencing
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NPV	Negative Predictive Value
OMIM	Online Mendelian Inheritance in Man
PCR	Polymerase Chain Reaction
PIR	Protein Information Resource
PPV	Positive Predictive Value
RNA	Ribonucleic Acid
SBS	Sequencing by Synthesis

SIB	Swiss Institute of Bioinformatics
SMOTE	Synthetic Minority Over-sampling TEchnique
SNP	Single Nucleotide Polymorphism
SV	Structural Variations
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
TM	*Text Mining*
TN	True-Negatives
TNR	True Negative Rate
TP	True-Positives
TPR	True Positive Rate
TXT	Text
UniParc	UniProt Archive
UniProt	Universal Protein Resource
UniProtKB	UniProt Knowledge Base
UniRef	UniProt Reference Clusters
VUS	Variant of Unknown Significance
WES	Whole Exome Sequencing
WGS	Whole Genome Sequencing

Context and Motivation

NGS is a technology that produces extremely large volumes of data. Sequencing of individuals is not only used for scientific studies, but also for the study of diseases. The existence of a particular mutation may increase the predisposition for a particular type of disease [3]. The use of bioinformatics tools and clinical databases allows researchers to understand the functional impact of detected variants. However, the magnitude of information aggregated in such clinical databases often baffles researchers, making it extremely challenging to perform definitive calls on functional impact.

Text Mining is a new and exciting area of computer science research that tries to solve the crisis of information overload by combining techniques from data mining, Machine Learning, natural language processing, information retrieval, and knowledge management [4]. Text Mining techniques help to reveal patterns and relationships in large volumes of textual content that are not visible to the naked eye. Using Text Mining techniques, it is possible to save time and resources: the process can be automated and the results from a Text Mining model can be consistently derived and applied to solve specific problems [5].

In the past, it was often hard to extract important insight from large volumes of text. These tasks required the use of complex modelling and programming tools that require computationally expensive resources. Now, with the advent of Text Mining techniques, we can save time and resources [4]. These types of processes can be automated, and the result of a Text Mining model can be applied in solving specific problems.

Text analysis commonly uses Machine Learning techniques such as clustering, classification, association rules and predictive modelling to discern meaning and relationships in the underlying content [6]. In this way, the goal of Text Mining is to derive implicit knowledge that hides in unstructured text and present it in an explicit form [7]. Over the years, there has been an explosion of tools that allow the mining of texts in biomedicine as well as other areas.

By using a tool for the annotation of genomic variants (previously developed in the group) we have been able to aggregate information from several xix

biological/clinical databases in a personalized way. However, the collected information from these databases is not always easy to read, as it often requires manual curation to understand, for example, whether a variant is disease-related or not. With advanced Text Mining applied to clinical unstructured texts retrieved from clinical databases, it will be possible to select only the most relevant data according to a given disease/phenotype.

Objectives

The annotation of genomic variants associated with diseases has become an accessible task through NGS. This technology produces large amounts of data that through bioinformatic tools and clinical databases, such as ClinVar, OMIM, UniProt and dbSNP/dbVar, allow to understand the functional impact of the detected variants. However, the magnitude of the information present in the clinical databases does not always allow the researchers an objective view, making it difficult to perceive the functional impact of such variants.
In light of this, the aim of this Thesis was to develop a tool, combining Text Mining and Machine Learning approaches that allows the extraction of information from the clinical description associated with annotated genomic variants in the widely used database Online Mendelian Inheritance in Man (OMIM). By selecting the most relevant information, we expect to predict the pathogenicity of a genomic variant, with a given certainty. This tool will allow the user to narrow down the amount of clinical information collected from public databases and thus enable relevant genomic variant data selection.

General Introduction

Since the completion of the human genome project in 2003, sequencing technologies have been developing exponentially. The possibility of sequencing the entire genome – Whole Genome Sequencing (WGS) [8] or just the coding region of the genome – Whole Exome sequencing (WES) [9] in a quick and inexpensive way is now possible through Next-Generation Sequencing [10].

One of the main concerns of the NGS is the volume of data produced in a single run. Storage and bioinformatic analysis have become a constraint as the ability to interpret and respond to all biological issues from WGS and WES is not an easy task.

The resulting data from sequencing should be analysed using bioinformatics tools, such as programs and biological databases, to identify and annotate variants present in the DNA sequence. The interpretation of the genetic variants allows to understand the relation between a variant present in an individual and a certain disease/phenotype.

1. Sequencing DNA technologies and techniques

1.1 Sequencing DNA technologies

In 1977 a method of DNA sequencing was developed by Frederick Sanger, based on chain termination or as commonly referred to as Sanger's sequencing [11]. In addition to Sanger, Walter Gilbert developed a method of DNA sequencing involving the chemical modification of DNA. Due to its efficiency the Sanger's method was considered the gold standard for nucleic acid sequencing [12] and was adopted as the primary technology in the "first generation" sequencing [13].

With the advent of the Sanger method, DNA sequencing has increased and in 1987 the company Applied Biosystems launched the first automatic sequencing machine (AB370) [13]. This launch marks the beginning of a new era in sequencing that allowed the development of technologies that facilitated the completion of The Human Genome Project in 2001 [14].

The demand for faster, more accessible and labour-intensive technologies has led to the emergence of increasingly capable sequencing tools that have accelerated the appearance of Next-Generation Sequencing. The **Figure 1** shown a timeline with the years in which a particular NGS platform/instrument was introduced in the market.

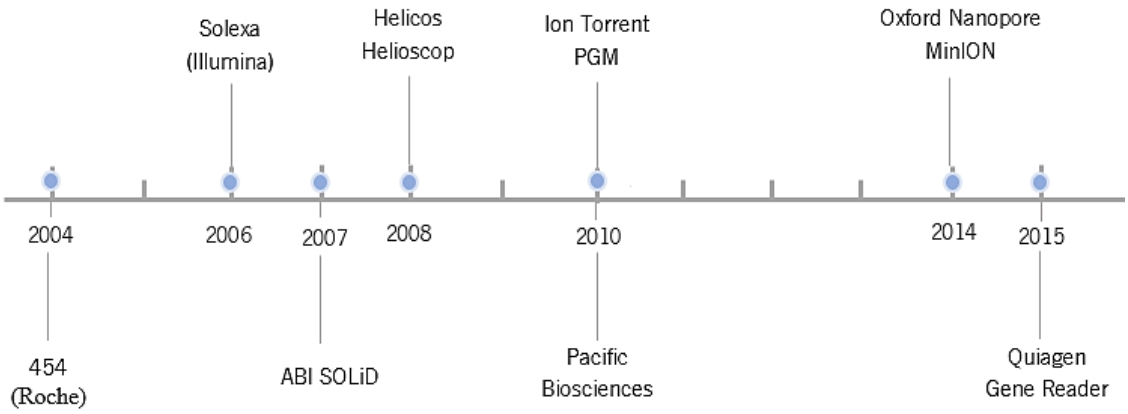


Figure 1 – Timeline with NGS platform / instrument developed over the years.

The completion of the human genome project allowed the emergence of the first line of sequencers not based on the Sanger method. This increased the volume of data generated, decreased cost-per-base and improved performance in terms of efficiency and accuracy, making DNA sequencing faster and massively parallel.

Shortly after the 454 launch, other companies began investing in the development of sequencing platforms. In 2007 the company Illumina developed a new method of sequencing called SBS (Sequencing by Synthesis). Since then the Illumina platform has been adopted as the standard for many applications in the field of genomics.

There are some errors associated with the Illumina platform, such as the decrease in the quality of the bases throughout the reads and the substitution of bases. The use of fluorescence makes sequencing less stable because the fluorescent signal used deteriorates as the reads length increases.

With technological development, new and more advanced NGS technologies have emerged. Ion Torrent created by the company Thermo Fisher is an NGS system that uses semiconductor chips. The differentiating character of this

system is that, unlike previously developed systems, it does not use fluorescence/ luminescence.

This new technology has been given the name PosLight technology [17] and offers greater speed and scalability and lower cost compared to light based systems. Ion Torrent has some associated problems, mainly in the sequencing of homopolymer regions (repetitive regions, rich in AAs and TTs). Compared to the Illumina platform, the Ion Torrent cost per reaction is lower but the error rate is much higher [15].

Platforms like 454, Illumina, SOLiD and Ion Torrent generate reads smaller than 500 base-pairs (bp), called short-reads. The use of short-reads makes it difficult to assemble complex genome. In order to solve this problem, the third generation of sequencing appeared, which enabled the sequencing of long reads.

The development of alternative sequencing technologies has experienced an exponential growth in recent years, where the main goal is offering technologies which are faster and more economical. Depending on the biological question in hand, different technologies can be selected, which in turn leads to choosing the most pertinent technique and bioinformatical data analysis method.

The third generation of sequencing is the most desirable for speed and reduction of error rate. Sample preparation is very fast, since a PCR step is not necessary, thus reducing the preparation time and the risk of bias and errors caused by the PCR technique. The runs associated with the samples can be made in just one day and the average reads length is 1300 bp which is higher than any existing second-generation technology.

One of the technologies most used in the third-generation is Nanopore created by Oxford Nanopore Technologies. Thus, in 2014 the Nanopore sequencer was developed [17], which uses the detection system based on membrane immobilized biopores with a diameter in the nanoscale [18]. Detection of nucleotides is done by analysing the variation of the electric potential in the

membrane, which is altered according to the passage of the fragmented DNA, as shown in **Figure 2**.

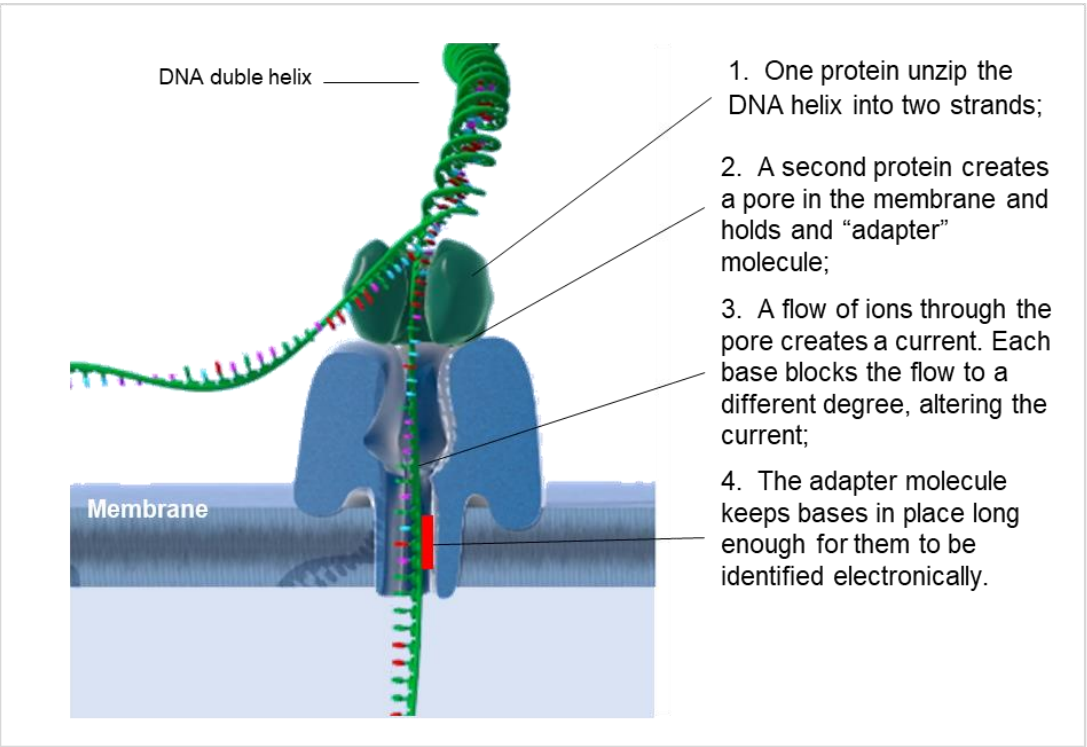


Figure 2 - Schematic figure of Nanopore sequencer working. Adapted from [18],[19].

Compared to other technologies Nanopore can reach the longest reads length (> 5kbp) with a speed of 1 bp/nanosecond. The use of enzymes makes this technology less temperature sensitive throughout the sequencing reaction and good results can be achieved more easily and efficiently. The development of these technologies has led to an increase in the DNA sequencing of many organisms, thus increasing the demand for research and development of more efficient and faster laboratory NGS techniques. The most commonly used NGS techniques are Targeted Sequencing (Panels), Whole-Exome Sequencing (WES), Whole-Genome Sequencing (WGS) and *de novo* sequencing. In the following section the NGS techniques mentioned above are described, explained and compared [13].

1.2 Sequencing DNA techniques

1.2.1 Target Sequencing (Panels)

Target Sequencing allows the isolated sequencing of a panel of genes or a region of the genome containing specific regions of interest. This technique allows sequencing with high coverage levels and easier identification of genetic variants [19].

The choice of panels is generally made considering prior or suspected knowledge that these set of genes/regions contain some association with the disease or phenotype under study.

This technique saves time and money (**Table 1**) since the investigation and analysis of the data is done considering only one or multiple areas of interest in the genome. This focused sequencing increases the level of coverage, allowing the identification of possible rare genetic variations that would be difficult to identify with more comprehensive NGS techniques such as WGS. Target sequencing provides a set of accurate and easy-to-interpret results in terms of volume than most NGS techniques.

Table 1 - Differences between NGS techniques.

Techniques	Cost	Data volume	Biological Driven
Targeted sequencing	$	+	+ + +
WES	$$	+ +	+
WGS	$$$	+ + +	-

Sequencing of gene panels is one of the most commonly used NGS techniques in clinical terms. Despite the wide use, this technique has some limitations, the main one being off-target enrichment, which is caused by the similarity of sequences between distinct zones of the genome due to, for example, events of genomic duplication during evolution. These particularities of the genome hinder the exclusive binding of primers to the target region, and binding of primers to regions completely outside the zone of interest may occur. For target sequencing it is recommended a high coverage of the region of interest, since sufficient coverage is crucial for the identification of genetic variants.

1.2.2 Whole Exome Sequencing (WES)

WES is the NGS technique that allows the study of the coding regions (exons) in the genome [9]. The exome is only 2% of the genome, however, 85% of variants related with genetic diseases are found in the exome. As one of the NGS techniques used in clinical approaches, WES has quickly become one of the main tools for the study of genetic causes of Mendelian diseases [20].

Being considered a directed NGS technique, WES has limitations also found in target sequencing. Enrichment outside the target hinders the isolation and sequencing of the exome. Thus, a significant portion of the readings obtained come from outside the previously defined target regions, for example, intron readings, intergenic regions and mitochondrial DNA [20].

Uneven coverage of reads on the target exome contributes to low coverage in many regions, which may result in missed variant calls. Regions with low coverage, high guanine-cytosine (GC) content, repetitive elements and segmental duplications are some of the limitations that hamper the downstream analysis of WES data [21].

WES produces a smaller set of data compared to WGS, since the sequencing of the entire genome results in an extremely large data volume. Compared to target sequencing, WES yields a bigger data set. Inherent to data set size, target sequencing and WES are less bioinformatically challenging when compared with WGS. Whole Genome Sequencing constitutes the final approach to detect all variants present in the genome of a patient in a single experiment, commonly used when target and WES have failed to reveal any relevant variants.

One of the limitations transverse to directed NGS techniques, as mentioned previously, is GC content. The amount of GC present in a DNA sequence is considered both a benefit and a detriment in molecular biology [22]. The high GC content may reduce the chance of mutation of A and T, which are more likely to create stop codons (TAA, TGA or TAG), which could modify protein synthesis. However, high GC content also has an associated downside because sites with high amounts of GC can generate CpG, *i.e.* regions of DNA where cytosine is followed by a guanine [23] which are prone to mutation [22].

The guanine-cytosine pair is linked by three hydrogen bonds, which are more stable than the two-hydrogen bonded adenine-thymine pair. This gives GC a higher molecular stability, to note, however, that hydrogen bonds are not the only ones responsible for GC stability [22]. Stacking bases through the hydrophobic interactions between two or more consecutive bases that minimize contact with water gives stability to the DNA molecule [24]. Thus, the stacking forces stabilize the double helix almost as much as the hydrogen bonds.

The stability of the DNA molecule when GC content is high hinders denaturation of the DNA strand during PCR. The melting temperature (temperature at which the double strand of DNA unfolds and separates into two single strands) used in the PCR depends on both the chain length and the sequence composition. The high GC content indicates a higher melting temperature and a more stable chain. The possible solution to this problem would be to increase the temperature of melting, however this compromises the other components, which with high temperatures disintegrate, decreasing the PCR product.

In conclusion, the low efficiency in the hybridization of the primers with the target sequence is due to the rich GC content [25]. The regions with less coverage are constituted by a greater amount of GC being these the promoter regions and the first exon of many genes [26]. This type of problems leads to a mistrust in the results of sequencing, and it is necessary to improve the techniques since the results are intended for diagnosis.

1.2.3 Whole Genome Sequencing (WGS)

In contrast with directed NGS approaches, WGS allows sequencing of the entire genome of an organism, without limiting or focusing on a region or panel of genes. It is considered the most comprehensive technique since it provides an overview of the whole genome.

Technological advances and a decrease in the cost of DNA sequencing have made WGS a reality. The WGS is a revolutionary option, which allows through the sequencing of the integral genome to identify genetic variants. The identification of variants facilitates the diagnosis of diseases with a genetic component, allows a better understanding of mechanisms behind many diseases and leads to the evolution of the concept of "personalized medicine" [8].

Compared to targeted NGS techniques, WGS could analyse different types of genetic variants in both coding and non-coding regions. Uniform coverage of reads is essential for good sequencing results. Contrary to targeted NGS techniques, WGS provides good results, since it is not necessary to develop primers to flank the region of interest, there is no possibility of off-target capture.

The cost of sequencing, storing and analysing a region with high depth is much lower in targeted techniques, the volume of data relative to WGS is also much lower (**Table 1**) [27]. The volume of data generated by WGS is enormous and there is still no biological answer to all the questions raised by the information contained throughout the genome sequence, which is the main disadvantage of this technique [27].

Targeted techniques have a lower cost than WGS, making them often more commonly used approaches in projects where financial capacity is lower. For some research all the information produced by WGS is not required, making targeted sequencing preferential.

An important consideration is that generating the data is only a fraction of the total cost and does not consider the costs associated with storing, analysing and interpreting data. As described in **Table 1**, targeted sequencing, compared to WES and WGS, is more economical and produces smaller volumes of data. In biological terms, it is more driven because it focuses on a panel of genes that can be associated with a certain disease/condition. In WGS, sequencing is applied to the entire genome, thus this technique is less biologically driven than target sequencing or WES.

1.2.4 *De novo* sequencing

De novo sequencing is an NGS technique with a different approach to previous the techniques, since a new genome is sequenced, without having a reference genome for alignment. This technique provides useful information for mapping genomes of new organisms or for completing genomes of known organisms [28], [29].

The process of mapping reads is complex, since there is no reference genome to which the reads can be aligned, they need to be assembled as contigs. The

size and continuity of contigs increases data quality and confidence in the sequenced genome. *De novo* sequencing will not receive great emphasis in this thesis, since it is a technique that is based on the discovery of new genomes and not in the study of the existing ones.

To conclude, there are currently several platforms that allow the application of different techniques, with their application depending on the biological context of the problem (**Table 1**)**.**

2. NGS Applications

Throughout the last decade there have been projects related to the sequencing and interpretation of the genome of many organisms. The Human Genome Project (HGP) [14], [30], concluded in 2003, led to the complete sequencing of the human genome and with it came an enormous amount of information hitherto unknown.

After the HGP, many projects followed, such as the 1000 Genomes Project [31], [32] which analysed 1000 anonymous individuals from different ethnic groups to find most of the genetic variants that have a frequency of at least 1% in the populations studied. International collaborations have enabled the creation of a catalogue of common genetic variants that are related to human diseases, carried out by the HapMap Project (short for 'haplotype map') [33].

As previously mentioned, with the advent of NGS, sequencing of the entire genome or part of it has become one of the main approaches in research and diagnosis of genetic diseases. NGS is the most widely used tool for detecting variants within the genome of any individual, yet with the amount of variants an individual contains in their genome, finding a variant related to a disease is hard work [10].

The use of NGS in the clinical area has been increasing, especially in the study of several diseases, one of which cancer. According to the World Health Organization, by 2015 cancer accounted for one in six deaths with an estimated number of deaths of 8.8 million [34]. Despite all efforts, cancer remains the second leading cause of death worldwide [34]. The use of NGS in the clinical

area, for identification of genetic variants, especially in the field of cancer, has a very extensive potential. This can lead to a diagnosis, guidance, and counselling of the patient being the genetic information useful in identifying family members who may also be at risk of developing this disease.

Detection of the variants that exist between the genome under study and the reference genome is a so-called variant calling process. The variant is found where there is a difference of one or more nucleotides between the genome under analysis and reference genome. The types of variants that can be found in a genome are described in the following section.

3. Genetic Variations

Sequencing enriches understanding not only of the genome sequence, but also of the genetic variations that occur in it. However, not all genetic variations are associated with disease: on average, each person is likely to carry approximately 250-300 variants of loss of function (LoF) in annotated genes and 50 to 100 variants previously involved in hereditary disorders [31]. In fact, genetic variations occur naturally in the human genome and that is what makes each individual unique.

Regardless of the molecular mechanisms or processes that generate the genetic variations, they can be broadly classified as germline or somatic. Germline variations occur in the germ cells of an individual and as they are inherited from the parents, all cells in the body will have this variant present. Somatic alterations are acquired throughout the life of an individual and are passed on to other cells by cell division (mitosis). This type of variation is common, most of which do not contribute to any relevant change in the individual [35]. However, some may lead to a change in phenotype or to increased susceptibility to disease, such as cancer, in cases where cells acquire the ability to proliferate and invade other tissues [35].

Genetic variations can span from one base pair to one million base pairs. There are variations that only involve changing a single base pair such as single nucleotide polymorphisms (SNP), to a few base pairs such as insertions and

deletions (indels) and at the other extreme, involving megabases (Mb), copy number variations (CNVs), inversions and translocations.

3.1 Single Nucleotide Polymorphisms

SNPs are the most abundant type of genetic variation in the human genome and occur once in every 1000 nucleotides, in at least 1% of the population [35]. The final phase of The 1000 Genomes Project characterized a total of 88 million genetic variants, of which 84.7 million are SNPs [31], [32]. These SNPs can be located either in coding regions of the genome (in exons) or in non-coding regions of the genome, *i.e.* intronic/intragenic or in intergenic regions.

SNPs located in the coding region of a gene may alter the encoded protein sequence, as these SNPs may lead to the replacement of an amino acid, which may affect protein function. If this happens, the variant is considered non-synonymous. If the protein sequence is unchanged, this means that the variant is synonymous and likely does not impact the encoded protein function.

Studies have shown that an individual contains many variants with functional consequences ranging from beneficial to highly deleterious. In fact, Durbin *et al.* have suggested that an individual typically differs from the human reference genome between 10,000-11,000 non-synonymous sites and 10,000-12,000 synonymous sites [31]. SNPs located in non-coding regions account to approximately 90% of all known SNPs [36], [37].

Genome-Wide Association Studies (GWAS) have shown that such non-coding SNPs may potentially be associated with human disease. Unlike the variants present in the coding region, which may directly impact the encoded protein, there is still few information on the impact of non-coding variants. One possible answer to this conundrum is that these SNPs cause changes in gene expression levels instead of causing changes in protein function [38].

There are some studies, related to variants present in non-coding regions. This field of human genetics studies the expression quantitative trait loci (eQTLs). eQTLs are genomic loci that help regulate mRNA expression, which in turn is the central key to regulate the expression of multiple genes. The presence of genetic

variants, for example, SNPs can be found in eQTLs by altering the gene expression of one or more genes, which are regulated by these loci [39].

3.2. Structural Variations

In ample sense, structural variations (SV) can be defined as all genomic changes that are not simple base pair substitutions [40],[41]. This variation includes insertions, deletions, inversions, duplications and translocations of DNA sequences, and covers copy number differences, also known as copy number variants (CNVs). CNVs include duplications, deletions and rearrangements, and represent a significant part of our normal genetic variability, and occur in both coding and non-coding regions [42]. About 34 million short indels and 60,000 structural variants were detected at the conclusion of the 1000 Genome Project [32].

Several structural genetic variables have been shown to be important both in phenotypic variability and susceptibility to diseases [43]. For example, the increased copy number of the CCL3L1 gene is associated with reduced susceptibility to HIV infection and progression to AIDS [44]. Likewise, individuals with fewer copies of the DEFB4 gene are at increased risk of developing colonic Crohn's disease [45].

Table 2 - Types of genetic variations. Adapted from [46].

Single Nucleotide Polymorphism (< 1 kb)	**Structural Variations (≥ 3 Mb)**
Small Insertion	Copy number variant (CNV)
	Indels
Small Inversion	Translocation
	Duplication
Small Duplication	Large-scale CNV (≥ 50Mb)
	Inversion

Defining genetic variants that are related to a specific disease can be valuable as it can lead to a definitive diagnosis, guidance and counselling of the patient and

this genetic information may be useful in identifying family members that may also be at risk of developing this disease. Defining genetic variants that are related to a specific disease can be valuable as it can lead to a definitive diagnosis, guidance and counselling of the patient and this genetic information may be useful in identifying family members that may also be at risk of developing this disease.

Sequencing as the main aspect in genomics has undergone major advances over the last few years, offering great benefits in several clinical areas, such as oncology. With the advent of next-generation sequencing technologies, the ability to sequence clinical samples in a massive and parallel fashion has enabled the discovery of new variants that may now be related to diseases/phenotypes.

4. Databases of Genetic Variants

While the NGS techniques develop rapidly, the volume of biological data accumulates, leading to an increasing number of databases in response to the amount of data generated. These advances have facilitated the discovery of numerous genetic variants that may or may not be related to certain genetic disorders.

Variations in the genome can range from the alteration of a single nucleotide to structural variants, as described in previous chapter (‘Genetic Variation’) in this thesis. As reported in the 1000 Genome Project [31], [32] it is estimated that an individual has carries between 10,000-11,000 variants that cause changes in the protein sequence, which can lead to highly deleterious consequences. In addition, it is estimated that an individual has between 10,000-12,000 synonymous variants that have no functional consequences.

The ability to interpret the variants present in the genome and what they represent in terms of function is an ongoing goal, with particular impact in disease. In fact, the clinical interpretation of the variants found in a given patient or family member can improve diagnostic efficiency and decision during treatment [47]. Moreover, misinterpretation of variants may affect clinical interpretation. Thus, databases collecting information on genetic variables found in the genome of a given

individual have a key role in clinics. Such biological/clinical databases can be cured at different levels. In fact, a good data curation is fundamental for the credibility of the data.

Biological databases can be curated by specialists as is the case of RefSeq [48], or can be curated in a community way, where they are curated collectively in collaboration with researchers, such as GeneWiki [49]. The range of biological data available can also vary from database to database. More comprehensive databases include data from several species, such as: GenBank [50] that provides nucleotide sequences for a large number of species and European Molecular Biology Laboratory (EMBL) [51], where DNA sequences, protein and related molecular information are available. These two databases are publicly available facilitating the exchange of data.

In addition to the databases mentioned above, there are databases that contain only human genome information. This information may be related to DNA, RNA, proteins and include clinical information that relates genetic variants to diseases /phenotypes. There are databases for each type of biological data, therefore choosing a database depends entirely on the data one is working with (**Table 3**).

Table 3 – Examples of Human-related biological databases.

Category	Name	Brief description	References
DNA	dbSNP	Database of Single Nucleotide Polymorphisms	[52]
	dbVar	Database of Genomic Structural Variation	[52]
	1000 Genomes	A deep catalog of human genetic variation	[53]
RNA	DARNED	Database of RNa EDiting in humans	[54]
Protein	UniProt	Universal protein resource	[55]
Disease	COSMIC	Catalog Of Somatic Mutations In Cancer	[56]
	OMIM	Online Mendelian Inheritance in Man	[57]
Clinical Information	ClinVar	Provides the relations between variants and phenotypes from a clinical point of view.	[58]

In this Chapter, the databases UniProt, OMIM, dbSNP, dbVar and ClinVar will be further detailed, with particular focus on ClinVar, as it will be the main database used during this Thesis.

4.1 UniProt: Universal Protein Resource

UniProt is produced by the UniProt Consortium, formed by the Swiss Institute of Bioinformatics (SIB), the European Bioinformatics Institute (EBI) and the Protein Information Resource (PIR). UniProt provides detailed information on the function of proteins, interactions, pathways, relationships with diseases and other areas of biological interest. UniProt Knowledge Base (UniProtKB) is considered the central source of UniProt and consists of two main sections, UniProtKB/Swiss-Prot that is manually annotated and UniProt/TrEMBL automatically annotated. Therefore, the UniProt comprises three databases: UniProtKB, UniRef and UniParc (**Figure 3**).

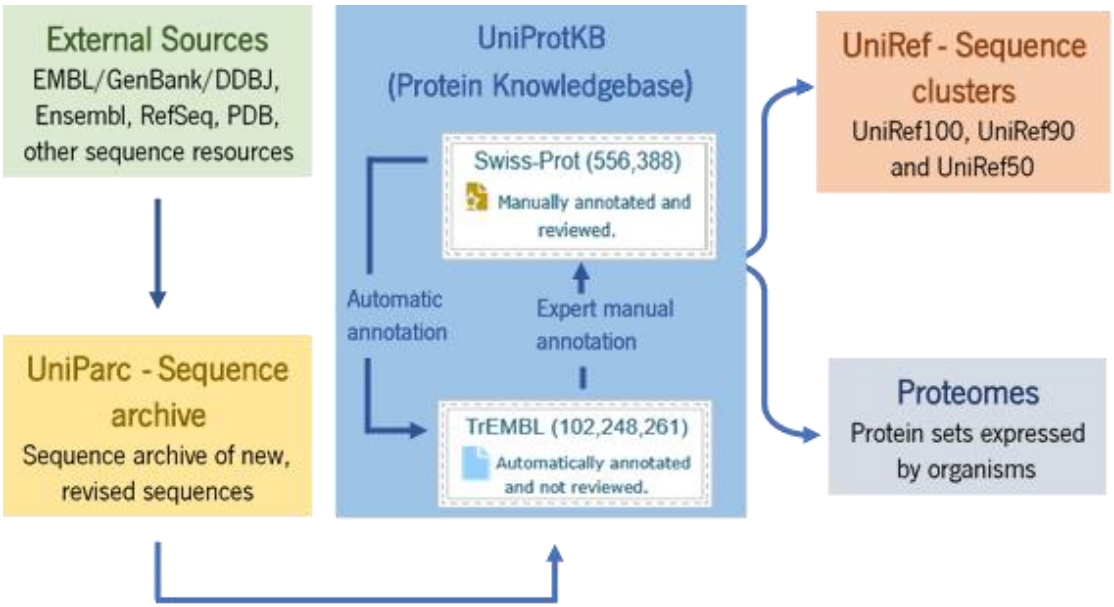


Figure 3 - Source and flow of data for UniProt's component Databases. Adapted from [59].

4.1.1 UniProtKB/Swiss-Prot

UniProtKB/Swiss-Prot contains the protein sequence information annotated manually and non-redundantly [59]. The manual annotation is made by an expert

curator and provides a critical review through experimental data for each protein and protein sequence.

In the section Swiss-Prot the curators obtain the information through literature and computational analysis [55]. The information extracted from the literature includes the name of the proteins, the ID, comments on the function, protein-protein interactions, pathways, location, among others.

4.1.2 UniProtKB/TrEMBL

UniProtKB/TrEMBL provides the computer records based on automatic annotation (unreviewed). The protein sequences present in this section derive from the translation of coding sequences (CDS) directly submitted to public databases such as EMBL, GenBank and DNA Data Bank of Japan (DDBJ) or other sources such as Ensembl. The entries in TrEMBL are manually annotated and then integrated into UniProtKB/Swiss-Prot [55].

4.1.3 UniProt Reference Clusters (UniRef)

UniRef consists of three databases, where the sequences are clustered by the percentage of identity between them consisting of three levels, UniRef100, UniRef90 and UniRef50 [59]. UniRef100 combines sequences and fragments that are 100% identical. UniRef90 and UniRef50 cluster the sequences with identity percentages of 90% or 50% respectively.

4.1.4 UniProt Archive (UniParc)

UniProt Archive (UniParc) was created to include all data and sequences of proteins accessible in public databases. When a given sequence exists in multiple databases it creates redundant information. UniParc memorizes each sequence only once and assigns it a unique UniParc identifier. Sequences are treated as text characters and only those that are 100% identical are mixed without regard to species. All sequences can be traced back to their original database because UniParc has cross-references to several source databases [55]. There may be proteins with the same sequence but having different

functions depending on the species or another variant, so the UniParc records are not annotated because the annotation depends on the context [59].

5. OMIM: Central Bioinformatics Resource for Human Disease

Online Mendelian Inheritance in Man (OMIM) is a database that provides detailed and cured information on genes and associated genetic disorders. The focus is on hereditary genetic diseases, where genetic traits [60] are transmitted from generation to generation in a Mendelian way, *i.e.* an alteration in the DNA sequence occurring in a single gene [61].

The information present in OMIM arises from biomedical literature and the use of aimed research in PubMed allows to extract information on the entry of genes and phenotypes (**Figure 4**).

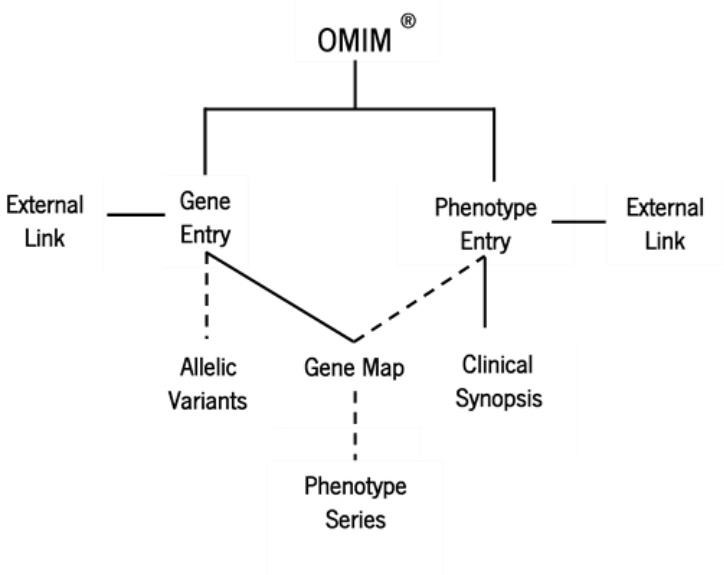


Figure 4 - Diagram of OMIM entries and content, adapted from [61]. Dashed lines indicate that not all genes have allelic variants, not all phenotypes are part of the genetic map and the mapped phenotypes are not all part of a phenotypic series.

The structure used by OMIM facilitates the search for information and depends on the entry. When the entry is a gene, OMIM, in addition to all relevant information, returns the allelic variants associated with the gene. It is possible to visualize a table with all variants, phenotypes and cross references with databases such as dbSNP, ExAC (Exome Aggregation Consortium) and ClinVar. If the entry is a phenotype, brief clinical description (clinical synopses) is

presented, containing information as inheritance, miscellaneous and molecular basis associated with phenotype/disease.

Recording of similar phenotypes / clinical manifestations may occur. Thus, genetic maps that combine cytogenetic localization and phenotype facilitate the creation of phenotypic series that demonstrate the genetic diversity of similar phenotypes in the genome [60].

Entries in OMIM receive an exclusive six-digit number, designated MIM number. Each digit or symbol of the MIM number is associated with a certain meaning, *i.e.*, where the first one digit indicates how the gene was inherited. Inheritance may be linked to the X chromosome, to a recessive, dominant or mitochondrial inheritance [61]. Another example, the allelic variants are given a MIM number followed by a decimal point and four unique variable digits [60].

Allelic variants are an important feature of OMIM, since they are at the basis of genetic-phenotypic relationships and some of them represent disease-causing mutations [61]. Each allelic variant in OMIM has a variant, title, mutation, and section number that provides a description of the mutation with data to prove and strengthen this information.

Regarding the information present in OMIM, gene entries, identified with an asterisk symbol (*), may include information such as gene structure, isoforms, expression and function [60]. Phenotype entries identified by a cardinal symbol (#), include description of clinical characteristics, patient and family reports, genetic discovers and links to clinical information. Phenotypes in OMIM include Mendelian disorders, phenotypic traits, susceptibility to a particular disease, among others [60].

OMIM is a source of essential information for the research areas related to health and biology. Being a database reviewed and curated in terms of scientific literature, OMIM becomes essential in understanding the relationship between variants, genes and diseases. The free text format for search makes it a versatile database, easy to query. Another advantage of OMIM is that all information presented is based on PubMed publications which is further reviewed by experts [62].

There are some disadvantages related with OMIM, one of them is the absence of clinical specificity [63] that is, inexistence of coherence in clinical terms. This problem arises because OMIM allows the text to be written in free and unstructured format [60]. Thus, the terms used for the same phenotype or disease vary depending on the user. In bioinformatic terms, written text in a free and unstructured format is also a complex problem that makes it difficult to extract and use the information contained in the database. This type of problem is one of the focuses of this thesis and will be discussed later.

6. dbSNP and dbVar

dbSNP is a database developed by the National Centre for Biotechnology Information (NCBI) containing all short (<50bp) sequence variations of nucleotides, and not just single nucleotide substitutions occurring with sufficient frequency in a population to be designated polymorphism. The data deposited in dbSNP can be from any organism, from anywhere in the genome, and include single-base nucleotide substitutions, small-scale multi-base deletions or insertions, and microsatellite repeats [64]. It also provides access to variations in the germline or somatic origin that are clinically significant.

The dbSNP has connection to several other databases, such as dbGaP (Database of Genotypes and Phenotypes) PubMed, ClinVar and dbVar. The variants submitted in dbSNP come from various sources, such as public research laboratories, private organizations and other databases. Each submitted variant receives a unique identifier called 'submitted SNP' ('ss #') number. There are cases where the same variant is submitted multiple times by different organizations, and it is necessary to aggregate this information into a single number, called 'reference SNP' (rs #) number.

dbVar is developed by NCBI and complements dbSNP. This database archives the genomic structural variations (≥ 50 bp), such as copy number variants (CNV), insertions, deletions, inversions, and translocations [65].

Table 4 - Distinguishing features of dbSNP and dbVar. Adapted from [1]**.**

Databases	dbSNP	dbVar
Variation Type	Small Variations (< 50bp): Single nucleotide variation (SNV); Short multi-nucleotide changes (MNV); Small deletions or insertions.	Large variations (≥ 50bp) Copy number Variants (CNV) Large deletions and insertions Inversions Translocations
Data Aggregation	Data by RS: Submitted SNP (ss) information Submitter contact and publications Variation Data – alleles, genotype, and frequency Experimental methods and conditions Genomic positions on different assembly versions ClinVar clinical assertions	Data by SV and SSV: Submitter contact and publications Method Genotype and Frequency Genomic positions on different assembly versions ClinVar clinical assertions
Linked Resource	ClinVar dbGaP Gene PubMed	Genome Nucleotide Protein Taxonomy

Structural variants are involved in complex human disorders and information such as location and type of variation can be accessed through dbVar. The interpretation of clinical relevance for a given variant can also be obtained through dbVar, since there is a cross-reference for ClinVar.

The quality of dbVar depends on the quality of the data provided by users, and it is important that all data is of high quality. There are consistency guidelines followed by dbVar that facilitate data verification. Data validation is done during the submission process, where errors such as inconsistent or invalid data can be verified. Errors considered serious interrupt the submission process and if necessary dbVar will contact the data submitter. For example, if the submitter's

variant location is easily detected as incorrect, the submission is returned, and the submitter is asked to verify and, if necessary, to correct the error.

There is a steady stream of information exchange from dbVar and another database storing genomic variants, the European Bioinformatics Institute's (EBI's) Database of Genomic Variants archive (DGVa). Together, these databases represent the largest archive of structural variation in the world. **Table 4** summarizes the distinguishing features of dbSNP and dbVar.

7. ClinVar

ClinVar is a database that includes clinical interpretation of genetic variants, which can be identified in any genomic site, without distinguishing its type, length or origin. Variants can be germline or of somatic source, with its identification being done through research or clinical cases. ClinVar available clinical interpretations are enriched with several databases, such as OMIM, GeneReviews, UniProt and dbSNP.

Each variant-phenotype interpretation added to ClinVar receives a number of accesses with the SCV prefix. The submissions have five data categories, which will be described below.

- **Submitter:**

  Variants interpretations can be added to ClinVar by organization and individuals.

- **Variation:**

  This is a fundamental component to represent the relationship variant-phenotype. The variation is defined as an alteration in a specific site or a combination of alterations in various sites of the genome. However, the data provided by some submitters is not able to establish a comparison. The free text that describes a variant is only accepted if it is connected to another public database.

Variants submitted in ClinVar are compared by the positions described in dbSNP and dbVAR. If the variant is already known, ClinVar assigns a variant calling identifier with the registry Reference ClinVar (RCV) or rs#. If the site of a variant is new, it is sent to dbSNP or dbVar to be submitted. One of the differences between ClinVar and dbSNP/dbVAR is that it does not create its own identifiers.

- **Phenotype**

Also designated as “condition”, it is quite similar to the ‘variation’ field. However, phenotype is represented as a unique concept or a group of concepts, being employed to report combinations of clinical characteristics. Authors/submitters are encouraged by ClinVar to send more phenotypical information such as identifiers from other databases, like MIM number, MeSH term or the identifier of human phenotype ontology (HPO). The interpretations are described in a free text format, however the usage of standardized concepts could help in the mapping of concepts. Currently, most submitters register diagnosis with a single term that has ample meaning.

- **Interpretation**

ClinVar receives an interpretation with the clinical meaning of a variant. Terminology used by ClinVar is recommended by the American College of Medical Genetics and Genomics (ACMG) [66], consisting of five terms:

i. pathogenic
ii. likely pathogenic
iii. uncertain significance
iv. likely benign
v. benign

To note that the term ‘likely’ is very broad, ACMG suggests that the term only be used when there is a 90% or more certainty that the variant is benign or causes a certain disease. Despite the fact that these terms do not cover all human phenotypes these five levels of classification remain relevant for Mendelian diseases [66].

- **Evidence**

This section contains the necessary details to support the interpretation of the clinical meaning of a variation-phenotype relationship, wherein the information is structured or summarized in a free format text, with an experimental or observational origin. To note that ClinVar does not allow evidence based purely on computational research, it can however be used as a complement to the other two information origins.

The evidence should include a description of the variant called and its biological context (genetic testing, tissue comparison cancer/normal, etc). The clinical interpretations of variants come from authors/ submitters but there is a revision done by external collaborators. **Table 5** shows the review status, number of gold stars and the description of each review status.

Table 5 - Classification according to the review status.

Review Status	Number of gold stars	Description
Practice guideline	Four	Practice guideline
Reviewed by expert panel	Three	Reviewed by expert panel (guest collaborators)
Criteria provided, multiple submitters, no conflicts	Two	Provided by two or more submitters with assertion criteria (the variants are classified according to the guidelines of the ACMG)
Criteria provided, conflicting interpretations	One	Multiply submitter, provided the same assertion criteria but with conflicts in interpretations
Criteria provided, single submitter	One	A submitter provided an interpretation with assertion criteria
No assertion for the individual variant	None	The allele was not included in any interpretation. It was only submitted as a component of a haplotype or a genotype
No assertion criteria provided	None	The allele was included in a presentation with an interpretation but without assertion criteria.
No assertion provided	None	The allele was included in a presentation but without interpretation

8. Comparison between databases

The annotation of genetic variants is essential to comprehend the knowledge associated with sequencing results derived from, for example, patient samples. Databases such as those previously described, compile annotation information, facilitating the flux of knowledge between organizations and individuals. The lack of standardized writing and formatting rules, often observed in clinical databases, limit the use of this information with bioinformatics tools. In fact, annotation is done traditionally in free text format which is easily readable by humans but presents obvious disadvantages when reading is done computationally.

Using the previously described databases as an example, OMIM is considered a non-classifying database since information appears in large text blocks, without standardized writing and formatting, making data extraction computationally complex. Databases such as UniProt and ClinVar are considered classifying databases since the information they possess is succinct and usually stored in table format, facilitating computational reading. While ClinVar provides structured information on genetic variants that are known be potentially pathogenic or benign with an associated phenotype/disease, OMIM provides useful clinical descriptions in an unstructured format. While ClinVar information allows computational parsing, the same is not valid for OMIM texts. Hence, the relevance of the work described in this Thesis, that is expected to extract relevant clinical information from unstructured texts as those provided by OMIM, using corresponding information from ClinVar to create and validate our tool.

9. *Text Mining*

The evolution of NGS technologies and technologies has allowed areas such as health and disease-related research to increase exponentially. The ability to sequence an entire genome (WGS) or regions of the genome (WES and target-sequencing) allowed the NGS to be integrated as a diagnostic tool for screening of variants present in the genome that may be related to a given phenotype/disease.

The volume of data resulting from NGS techniques is enormous, compared to basic techniques such as the Sanger sequencing. The large volume of data, despite the underlying disadvantages, such as the need for high capacity and computational storage, the complexity in the analysis and interpretation of data, allowed the increase of biomedical knowledge.

Biomedical texts provide a valuable source of knowledge in biomedical research. In the case of cancer there are approximately 3,605,848 publications in PubMed, where the word cancer appears in the title or abstract. The enormous biomedical knowledge available in publications and databases there is a constant interest in methods capable of identifying, extracting, managing and exploiting this knowledge and discovering new and/or hidden knowledge. The biomedical information currently available makes it impossible for analysis and interpretation to be done manually. In this way, the concept of *Text Mining* emerges, which can be characterized as the process of analysis of unstructured and ambiguous texts to extract useful information of high quality and relevance, thus making the text more accessible in research terms.

Text Mining is similar to data mining, however the tools used in data mining are designed to handle with structured data available in databases. *Text Mining* works with unstructured data such as e-mail, text documents, etc The main purpose of *Text Mining* is to transform text (unstructured texts) into text (structured) for analysis, using natural language processing (NLP) methods [67].

Text Mining is a multidisciplinary field that incorporates many areas such as information retrieval, information extraction, the natural language processing and data mining [67]. *Text Mining* encompasses several areas, which interconnect to extract useful information from unstructured texts.

9.1 *Text Mining* Areas

- Information Retrieval (IR)

IR is usually the first step to handle textual data from a large collection of important documents. The IR system is used in the biomedical field to assist researchers in finding search-related articles, in search engines such as PubMed, and Google. IR systems allow restricting the set of

documents that are relevant to a specific problem, significantly accelerating the analysis [68].

- Information Extraction (IE)

It is the process of automatically extracting structured information from unstructured text documents. An IE system involves the identification of entities such as names of people, companies and location, attributes and relationships between entities [67]. The system does this through pattern recognition.

- **Natural Language Processing (NLP)**

Natural language processing is one of the most challenging problems of artificial intelligence NLP looks for the ability of the computer to understand the natural language as do humans, thus having the ability to perceive the meaning of a sentence or document [69].

Natural language generation (NLG) ensures that the generated text is grammatically correct. Most NLG systems ensure that grammatical rules, such as agreement of the subject's verb, are obeyed and it is possible to decide how to organize sentences, paragraphs in a coherent way.

- **Data mining (DM)**

Data mining can be described as looking for patterns in text and by extracting hidden information, previously unknown and useful data. The main goal of the data mining process is to extract information from a dataset and transform it into an understandable structure for later use. Data must be first converted and next transformed into a format that allows easy knowledge extraction [70]. Different steps of data transformation may be applied depending on the original text format retrieved from a given database [71]. **Figure 5** shows the steps in the *Text Mining* process. These steps will be described, exemplifying some of them with information taken from ClinVar.

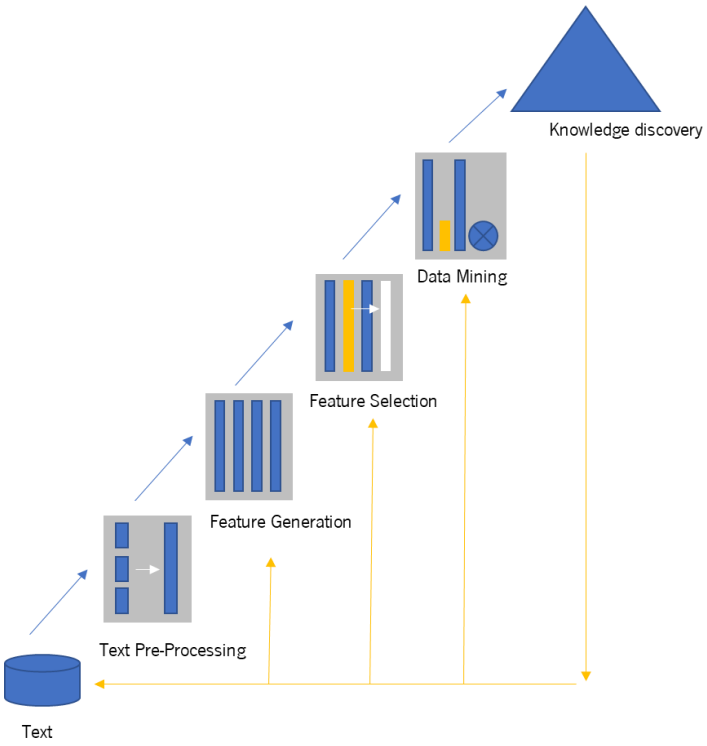


Figure 5 - Process of *Text Mining*. Adapted from [70].

9.2 Process of *Text Mining*

9.2.1 Text Pre-Processing

During processing, the text is pre-processed to allow a superior application of the various *Text Mining* techniques. These will be described below.

- Case Folding

One of the first data treatments to be performed is to convert all letters to uppercase or lowercase. Case folding is a procedure intends to standardize the words so that they can be identified in the text in uppercase or lowercase letters in the future, allowing a faster character comparison process. For example:

Database extracted data:

'somatic heterozygous G-to-A transition in the NRAS gene, resulting in a gly12-to-asp (G12D) substitution';

After Case Folding treatment of extracted data:

‘**somatic heterozygous g-to-a transition in the nras gene, resulting in a gly12-to-asp (g12d) substitution’.**

- Tokenization and Removal of unwanted characters

This step removes unnecessary characters such as punctuation that does not provide additional information [70]. From this, all words will be separated by space. This is a complex process because it is necessary to understand the domain of the text and which characters may contain important information. In the case of biology, it is important to keep the ‘+’ and ‘-‘ characters, because in the DNA strand, ‘+’ refers to the sense strand and ‘-‘ to the antisense strand. For example:

Database extracted data:

‘somatic heterozygous G-to-A transition in the NRAS gene, resulting in a gly12-to-asp (G12D) substitution’;

After Tokenization:

‘somatic heterozygous G to A transition in the NRAS gene resulting in a gly12 to asp G12D substitution’.

- Stop Words Removal

In this step, words without meaning / utility, *i.e*. that do not provide additional information are removed [70]. Examples of these words are ‘a’, ‘is’, ‘of’. Usually, 40 to 50% of the total words in the text are removed in this step. For example:

Database extracted data:

‘somatic heterozygous G-to-A transition in the NRAS gene, resulting in a gly12-to-asp (G12D) substitution’;

After Stop Words Removal:

‘somatic heterozygous G-to-A transition NRAS gene, resulting gly12-to-asp (G12D) substitution’.

- Stemming

Stemming is a process that reduces derived words, *i.e.* that turns words into their root. For example, when stemming is applied to words such as “computer” and “computing”, these words will be reduced to "compute" as it is its root [70]. This process can reduce verbal plurals and conjugations and, thus, the learning model can classify a document correctly, also reducing the number of words evaluated. In this way, the high dimensionality of TM applications is reduced, making it possible to use less computer space and shorter machine execution time. However, stemming has several drawbacks. Sometimes words that have the same root may have different meanings: for example, the English terms “desert” and “dessert” are words whose root is "des" but have completely different meanings. In fact, if it is poorly handled, stemming algorithms can greatly impair the result of the analysis. The major risks involved in this process are under-stemming, over-stemming and mis-stemming. Under-stemming refers to when a suffix is not removed, or a smaller suffix was removed than it should. Over-stemming, unlike the previous one, is when the stemming procedure has removed more suffix than it should [70]. Mis-stemming occurs when the stemming takes part of the word because it detected a suffix that was not. Nevertheless, stemming is a powerful tool for improving the performance of TM, which should be used with parsimony and insight into the original text under analysis [72]. For example:

Database extracted data:

‘somatic heterozygous G-to-A transition in the NRAS gene, resulting in a gly12-to-asp (G12D) substitution’;

After Stemming:

‘somatic heterozygous G-to-A transition NRAS gene, result gly12-to-asp (G12D) substitution’.

- Named Entity Recognition (NER)

NER is a process of identifying terms that relate to an entity present in the text. Entities generally fall into predefined sets of categories, such as person, location, organization. In the case of biology-related texts, these categories can be proteins, DNA, RNA and cell lines or cell types. For example, cancer can be represented as a disease as well as an astronomical sign [73].

- Synonyms Handling

Synonyms handling is the process of replacing words in the text with synonyms so that all words having the same meaning are replaced by the same word. This step allows you to significantly reduce the number of terms in the dataset without changing the meaning of the text.

- Word Validation

This step aims to validate the words found by the algorithm by comparing them to words in a dictionary. There are dictionaries available online with common terms, mainly in English, and other dictionaries with more specific terms, for example in the field of medicine or biology.

9.2.2 Feature Generation

After the pre-processing and now, having the texts cleaner and without words and characters that can be difficult the computational analysis. The next step is transforming the texts in a format that the computer can ‘read’. Therefore, there are three best known methods for transforming the dataset.

- Bag of Words Model

This model is considered the simplest in the field of *Text Mining*, since it only involves the transformation of unstructured text into numerical vectors, *i.e.* the Bag of Words Model represents the text as a numerical vector, where each number represents a specific word in a set of texts/documents (corpus). The numbers are the frequency of the word in the text [74]. The model is literally

represented as a 'bag' of the words that are present in the corpus, disregarding the order of words and grammar. Thus, texts are converted into numerical vectors, so that each text is represented by a vector (line) in the features matrix.

Table 6 - Bag of Words Model.

	polymorphism	**increased**	**loss**
Text 0	0	1	1
Text 1	0	0	0
Text 2	0	0	1

Thus, each column represents a feature and each row a text/document. The value of each cell represents the number of times each word (represented by column) occurs in a specific text (represented by line).

- Bag of N-Grams Model:

A word is just considered a token (output from tokenization) and is called in *Text Mining* as a monogram. However, we already know that the Bag of Words model does not consider order of words. Therefore, so if we want to consider the order of words in the document / text, N-grams may be useful. Hence, an N-gram is basically a collection of words/tokens from a text such that these tokens are contiguous and occur in a sequence. A Monogram indicate a n-grams of order 1 (one word), Bi-grams is a n-grams with two words and Tri-grams a n-grams with 3 words. Of notice, the Bag of N-Grams model is hence just an extension of the Bag of Words model and we can use the n-grams as features [75]. Similarly, with Bag of Words model an example of an output from the Bag of N-grams model is:

Table 7 – N-grams using the Bag of N-Grams Model.

	no polymorphism *n=2, bi-gram*	**increased risk** *n=2, bi-gram*	**loss of function** *n=3, tri-gram*
Text 0	0	1	1
Text 1	0	0	0
Text 2	0	0	1

The Bag of N-grams model provides the features for our texts, where each feature consists of a monogram, bi-gram and a tri-gram represented as a sequence of one, two or three words and values represent how many times the N-Grams were present into our texts.

- TF-IDF Model

There are some problems adjacent to the Bag of Words models (Bag of Words Model and Bag of N-Grams Model), since both feature vectors are based on the frequency of words/terms, hence, some words/terms that may occur more frequently in all texts and may tend to overshadow other words/terms in the feature set. Therefore, the TF-IDF model tries to solve this problem by using a scaling or normalizing factor. The TF-IDF is widely used in TM and allows to measure the importance that a given word has in a document or in a set of documents (corpus). The TF-IDF stands for Term Frequency-Inverse Document Frequency, which uses a combination of two metrics namely:

- Term Frequency (TF), which is by definition the frequency in which a word occurs in the document or corpus. Therefore $\boldsymbol{TF(t,d)}$ is the number of times a term $\boldsymbol{t}$ occurs in a document $\boldsymbol{d}$ [74].

- Inverse Document Frequency (IDF), allows to reduce the weight of commonly used words and increases the weight of words that are not widely used throughout the corpus [74]. IDF can be calculated as follows:

$$\boldsymbol{IDF(t,d) = log_{10} \frac{N_d}{d_f(d,t)}}$$

In this formula, $\boldsymbol{N_d}$ is the total number of documents and $\boldsymbol{IDF(t,d)}$ is the number of documents $\boldsymbol{d}$ that contain the term $\boldsymbol{t}$. The $\mathbf{log_{10}}$ is used to ensure that low documents frequencies are not given too much weight [74]. Therefore, the TF-IDF formula can be defined as the product of the term frequency and the inverse document frequency:

$$\boldsymbol{TF - IDF(t,d) = TF(t,d) \times IDF(t,d)}$$

The TF-IDF calculations were performed with the R package *tidytext*. This R package has powerful functions to deal with specific format of text widely used in TM approaches. The *tidytext* format is a table with one-token-per-row, this means that each row in the table is a string from the 'list of strings' that was given by the tokenization step.

9.2.3 Feature Selection

Feature selection is also known as variable selection and is the process of selecting subsets of relevant characteristics that will be used in the construction of the prevision model. The central point of the feature generation is that the terms present in the bag-of-words are often redundant and irrelevant and therefore can be removed without loss of information [76]. Emphasizing that a relevant term may be redundant in the presence of other relevant terms with which it is correlated.

9.2.4 *Machine Learning* approaches

At this point, *Text Mining* merges with *Machine Learning* approaches that allow the extraction of knowledge from texts. Some of these techniques are described below.

- Unsupervised Learning:

Clustering is a data mining technique that makes a meaningful or useful grouping of terms that have similar characteristics automatically. The clustering technique defines classes and puts the terms in each class [74]. Clustering involves grouping data, using either agglomerative clustering or divisive clustering techniques, with the aim of minimizing the distance between objects within the same cluster and maximize the distance between objects of different clusters. It is primarily concerned with distance measures and clustering algorithms which calculate the difference between data and group them accordingly. Clustering is most commonly referred to as 'unsupervised learning technique', *i.e.* the technique learns the inherent structure of data without using explicitly-provided labels. Since no labels are provided, there is no specific way to compare model

performance in most unsupervised learning methods. The Unsupervised learning can be useful to automatically identify structure in data.

- Supervised Learning:

The Supervised Learning is performed in the context of classification, when we want to map input to output labels and in the context of regression, when we want to map input to a contiguous output [74]. Common algorithms in supervised learning include Logistic Regression, Naïve Bayes, Support Vector Machine, Artificial Neural Networks, Decision Tree and Random Forest. Therefore, in both regression and classification, the goal is to find specific relationships or structure in the input dataset that allows us to effectively produce correct predictions.

When constructing a supervised learning, the main considerations are the complexity of the model and the variance of bias. The complexity of the model refers to the complexity of the function you are trying to learn. The appropriate level of complexity of the model is usually determined by the nature of the training data. If the training dataset has only a small amount of data or if the data is not evenly distributed across different possible scenarios, one should choose a low complexity model. This is because a high-complexity model will likely overfit if used on a small number of data points. Overfitting refers to learning a function that fits your training data very well but does not generalize to other data points—in other words, you are strictly learning to produce your training data without learning the actual trend or structure in the data that leads to this output. Therefore, the classification is a classical technique of data mining based on *Machine Learning* and is used to classify each word in a set of texts. For the purpose the most relevant methods were:

- **Decision Tree:** The Decision Tree influences a wide area of *Machine Learning*, covering both classification and regression. A Decision Tree can be used to visually represent decisions made by the algorithm. A Decision Tree is drawn upside down with the root node at the top [74], [77]. Therefore, each node represents a feature on which the tree divides into branches. The branch where there are no more divisions is called sheet and is where the final decision is.

The real sets have many features and the importance of the features and the relationships between them can be visualized along the tree structure. Thus, this methodology is known as learning Decision Tree from date and trees where the leaves are numbers or concrete classification as for example, 'Benign' and 'Pathogenic' are called Classification trees, however if in the sheet the decisions foreseen are continuous values for example, the price of a house, are called regression trees.

Creating a Decision Tree involves deciding which features to choose and which decisions to use to split trees and especially know when to stop tree growth. Therefore, we must be pruning it so that the tree only considers the important features and does not grow indefinitely. There are two common techniques used to splitting the data for the several branches in the Decision Tree, divide and conquer and cost of a split.

After, the Decision Tree starts to be divided considering each feature in the training data. However, in real problems the number of features is large, and this results in an enormous number of divisions in the Decision Tree which creates a large tree. These trees are considered too complex and can lead to overfitting. Therefore, it is necessary we to know when to stop. One technique to do this is to set a minimum number of training entries to use on each sheet. For example, we can use a minimum of 10 samples to reach a certain decision and ignore any leaf that takes less than 10 samples. Another technique is to define the maximum depth of our model. The maximum depth refers to the length of the longest path of the root one of the leaves. The pruning is a method that allows us to increase the performance of the Decision Tree, removing branches which using low important features [74], [77]. Therefore, reduce the complexity of the tree, increase the predictive power of the model by reducing the overfitting. There are several pruning methods some more sophisticated than others, such as reduced error pruning and cost complexity pruning. The Decision Tree method is simple to understand, interpret and visualize, can handle both numerical and categorical data and can also handle with multi-output problems. However, the Decision Tree method could create over-complex trees that do not generalize the data well (overfitting), also create biased

trees if some classes dominate, hence it is recommended to balance the dataset prior to build the Decision Tree method.

- Random Forest: Random Forest is a practical *Machine Learning* supervised algorithm, even without hyperparameter tuning, it produces great results frequently. It is one of the most commonly used ML algorithms because it is simple and can be used for classification and regression problems [74], [77]. Basically, the Random Forest algorithm create several Decision Trees and merges them together to get a more accurate and stable prediction. Therefore, the Random Forest in classification is considered the building block of *Machine Learning* and will be explained below.

Random Forest adds randomness to the model, *i.e.* unlike Decision Trees, instead of searching for the most important feature when dividing a node, it searches for the best feature among one of the random subsets of features. This method considers also the feature importance, once measures the relative importance of each feature in the prediction. Therefore, several tools measure the importance of each feature considering how much the nodes of the Decision Trees that use this feature, reducing the noise in all Decision Trees. Considering the importance of features, we can decide which features we want to remove, since they do not provide enough for the prediction process. This is one of the key points in the Random Forest method, once the *Machine Learning* methods are more likely to be overfitting and vice-versa. Another important aspect in the Random Forest method is the hyperparameters that can be added to increase the predictive power of the method or to make it faster. Therefore, there are numerous advantages to using Random Forest method, as the ability to adjust the method to classification and regression data, and it is easy to visualize the relative importance of the input features. It is also considered a practical algorithm, once the standard hyperparameters generally produce a good predictive result, moreover the addition of hyperparameters are easy to understand.

However, a major problem in *Machine Learning* is the overfitting as mentioned earlier. The main limitation of Random Forest is that a large

number of trees can make the algorithm slow and inefficient for real-time predictions [74]. Moreover, are fast algorithms to train but very slow to create forecasts after being trained. In many real-world applications, the Random Florets algorithm is fast enough, but there are certainly situations where run-time performance is important and other approaches may be preferred.

- **Decision Tree** *vs***. Random Forest:** although Random Forest is considered a collection of Decision Trees there are some differences. Therefore, if the input of a Decision Tree is a set of training data with features and labels for each instance, the method will formulate a set of rules that will be used to make predictions. In comparison, the Random Forest method randomly selects instances and features to construct multiple Decision Trees and then calculates the average of the results. Another difference between these two ML methods with supervised is that deep Decision Trees can suffer from overfitting. However, Random Forest prevents overfitting most of the time by creating random subsets of features and constructing smaller Decision Trees using those subsets. Then combine these subtrees, yet this type of strategy is not always effective and makes computing slower, depending on how many trees our Random Forest creates.

As previous mentioned, the supervised Machine Learning methods allows us to calculate several performance measures to evaluate the prediction capacity. Therefore, we use the values in the confusion matrix that is the output object from the ML model (Table 8). Hence, if we had, a binary classification such as ‘Benign’ (positive class) and ‘Pathogenic’ (negative class), after built a ML model, the output object is a confusion matrix similar to Table 8 to have the prediction values.

9.2.4.1. Performance Evaluation Measures

Table 8 - Example of a Confusion Matrix.

	Predicted	
Actual	**Benign**	**Pathogenic**
Benign	True-Positive	False-Negative
Pathogenic	False-Positives	True-Negative

Considering the Confusion Matrix, there are four important terms:

- True-Positives (TP): the cases in which we predicted 'Benign' classification and the actual output was also 'Benign' classification;

- True-Negatives (TN): the cases in which we predicted 'Pathogenic' classification and the actual output was also 'Pathogenic' classification;

- False-Positives (FP): the cases in which we predicted 'Benign' classification and the actual output was 'Pathogenic' classification;

- False-Negatives (FN): the cases in which we predicted 'Pathogenic' classification and the actual output was 'Benign' classification.

The values in confusion matrix (Table 8) allows the calculation of the following performance measures:

- **Accuracy**: it is the ratio of number of correct predictions to the total number of input samples [77]. Therefore, the accuracy is calculated considering the average of the values across the 'main diagonal', *i.e.*:

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{True\ Positives} + \boldsymbol{True\ Negatives}}{\boldsymbol{Total\ number\ of\ samples}}$$

- **Precision** (Positive Predictive Value, PPV), for the 'Benign' classification (positive class) is defined by [77]:

$$\boldsymbol{Precision} = \frac{\boldsymbol{True\ Positives}}{\boldsymbol{True\ Positives} + \boldsymbol{False\ Positives}}$$

- **'Negative' precision** (Negative Predictive Value, NPV), for the 'Pathogenic' classification (negative class) is defined by:

$$\boldsymbol{'Negative' Precision} = \frac{\boldsymbol{True\ negatives}}{\boldsymbol{True\ Negatives\ } + \boldsymbol{False\ Negatives}}$$

- **Recall**, also known as True Positive Rate (TPR) for the 'Benign' classification is defined by the formula [77]:

$$\boldsymbol{Recall} = \frac{\boldsymbol{True\ Positives}}{\boldsymbol{True\ Positives} + \boldsymbol{False\ Negatives}}$$

- '**Negative' recall** (True Negative Rate, TNR) for the 'Pathogenic' classification is defined by the formula:

$$\boldsymbol{'Negative' Precision} = \frac{\boldsymbol{True\ negatives}}{\boldsymbol{True\ Negatives\ } + \boldsymbol{False\ Positives}}$$

- **F1-Score:**

Another measure to evaluate the performance of the models is the F1-score that combines the precision and recall using the harmonic mean. Therefore, the F-measure has an intuitive meaning [78]. F1-score measures the accuracy of your classifier, i.e. how many instances it sorts correctly. Furthermore, it measures the robustness of the model, i.e. it checks if the model does not lose a significant number of instances. Therefore, with high accuracy, but low recall, the classifier is extremely accurate, but loses a significant number of instances

that are difficult to classify, however, this is considered to be of little use. F1-Score is the harmonic mean of precision and recall, so when we optimize a rating model by increasing one we disadvantage the other and the harmonic average decreases rapidly [74]. However, the harmonic average is higher when accuracy and recall are the same. The following is the formula for F1-score:

$$\boldsymbol{F1 - score = \frac{2 \times precision \times recall}{2 \times (recall + precision)}}$$

As described above, we have also created an alternative formula focused on the 'negative' precision and 'negative' recall. The following would be the formula for the 'negative' F1-score:

$$\boldsymbol{'Negative'\ F1 - score = \frac{2 \times 'negative'\ precision \times 'negative'\ recall}{2 \times ('negative'\ recall + 'negative'\ precision)}}$$

- ROC curve and AUC:

Another possible measure for the performance of a given model is the 'Receiver Operating Characteristic' (ROC) curve and the associated statistic 'Area Under the ROC curve' (AUC). The ROC curve is a graphical plot representing the FPR in the *x* axis and the TPR in the *y* axis, thus showing the diagnostic ability of a given classifier system. For example, a perfect classifier, with a TPR of 1 and an FPR of 0, would fall in the top left corner of the ROC curve [74], [79]. The AUC is a directly-related statistical measurement that characterizes the performance of the model, with 1 being the maximum value. For example, a given classifier with an AUC of 0.95 would be considered very good. However, ROC curves and AUC are not ideal measurements when dealing with models built using imbalanced datasets [74], [79]. In fact, Precision and Recall measurements are considered to be more informative on a given model performance, when the original dataset is imbalanced [80].

9.2.5 Knowledge Discovery

The knowledge discovery is the result of the whole process of *Text Mining*. After all the steps in the *Text Mining* process, it is necessary to extract knowledge from this information. Knowledge can lead to new discoveries [81] and to the increase of previously acquired knowledge. Knowledge is not extracted by any algorithm, the computational part only serves to facilitate the extraction of information, not the interpretation of information. For the interpretation and understanding of the information resulting from the process of *Text Mining* human intervention is necessary.

Nam and Park [82] used *Text Mining* and discovered two pathways functionally involved in the predictor gene set, indicative of susceptibility to early-onset colorectal cancer, overcoming the lack of studies of colorectal cancer expression throughout the genome.

9.2.6 Hypothesis Generation

The generation of hypotheses is to obtain an unproven conclusion through information hidden in the text, while the discovery of knowledge means to extract innovative knowledge. *Text Mining* methods can facilitate the generation of biomedical hypotheses, suggesting new associations between diseases and genes.

The biomedical literature is essential to extract potential information to make biomedical inferences and generate new hypotheses. The generation of hypotheses is an important task in *Text Mining* and is increasingly used by biomedical researchers wishing to infer unfamiliar biomedical facts.

One of the best-known examples for generating hypotheses came when Swanson found a connection between fish oil and Raynaud's syndrome. Thus, it has been hypothesized that fish oil may be useful in reducing high blood viscosity and high platelet aggregation, attenuating the symptoms of Raynaud's Syndrome.

The creation of protein-protein ratio maps specific to Alzheimer's disease based on interaction and mining networks provided Li *et al.* [83] propose a new

hypothesis where diltiazem and quinidine can be investigated as candidate drugs for the treatment of Alzheimer's disease.

10. *Text Mining* software and tools

The development of software that allows the extraction of knowledge from unstructured texts, has had special attention over the years. The emergence of several software and tools that allow the execution of the *Text Mining* process are an asset in the areas of biomedical research where it is necessary to read and analyse several publications and large amount of texts provided by databases.

In terms of information retrieval (IR) systems, PubMed is one of the most well-known biomedical databases, as well as MEDLINE and scientific journals that provide a wealth of information across the web.

In the pre-processing domain, software such as Acromine provides a dictionary of abbreviations that is built automatically from all MEDLINE. BioLexicon represents the set of lexical information terms that improve *Text Mining* performance. This tool gathers terminologies from large data sources such as UniProtKb and NCBI. Since the identification of the entity is a fundamental part of the *Text Mining* process, the connection of tools/software to large bibliographic databases is extremely important, for which GENETAG, one of the most used tools in the field of *Text Mining*, compiles about 20,000 sentences from MEDLINE for the identification of the term gene/protein. The recognition of named entities (NER), also being a stage of pre-processing, allows the identification of specific terms such as genes, proteins, diseases, etc. There are software and entity recognition tools that use *Machine Learning* algorithms that facilitate entity search and entity-term relationship.

A few tools available, such as iHOP that provides quick and accurate summary information covering approximately 80,000 biological molecules that are automatically extracted from key phrases from millions of PubMed documents [84]. Thus, iHOP allows the researcher to explore a network of gene and protein interactions by navigating directly in published scientific literature. Instead of providing long lists of entire abstracts after searching through keywords, iHOP

retrieves and selects specific information about genes and proteins and summarizes their interactions and functions.

BioText- Quest [85] is a biomedical *Text Mining* or concept discovery system that provides services such as biomedical recognition of named entities, association of concepts, and generation of hypotheses. It was initially constructed with 1000 abstracts of MEDLINE randomly selected on the yeast theme, later the dataset was annotated manually, now including abbreviated denunciations, protein-protein interaction data and the relationship between entities related to the treatment of diseases.

11. Application of Biomedical *Text Mining* in cancer research

As a complex disease, cancer is related to many genes and proteins. Biomedical researchers are interested in extracting cancer-related genes and proteins from the literature to study cancer diagnosis, treatment and prevention. Chun *et al.* [86] developed a system of recognition of entities and relationships between prostate cancer and relevant genes. Deng *et al.* [87] used a *Text Mining* approach to identify genes related to prostate cancer as candidate genes and using the Online Mendelian Inheritance in Man (OMIM) database to verify them. Krallinger *et al.* has implemented two cancer-related *Text Mining* applications [88]. One was used to extract human genetic mutations from predefined types of cancer from literature, the other was particularly used for classification of breast cancer and classification of breast cancer genes. One of the important areas of the cancer research is risk assessment, which determines the probability of developing cancer, evaluating the available evidence, through research and studies already existent and through *Text Mining* it is possible to collect and extract information that allows a possible early detection, for the prevention and management of patients always with the aim of reducing and controlling the causes of cancer.

Despite these examples, there are currently no tools that can interpret unstructured texts from clinical databases to infer pathogenicity in a global manner. This gap in knowledge is expected to be filled by the tool developed with this Thesis, which has used OMIM unstructured clinical texts concerning ClinVar-classified genetic variants as input for a

combination of *Text Mining* and *Machine Learning* approaches. This tool is expected to help the user to narrow down the amount of clinical information collected from public databases, such as OMIM, and thus enable relevant genomic variant data selection.

Methods, Results and Discussion

The aim of this Thesis is to develop a *Text Mining* tool that allows the classification of a genomic variant as 'Benign' or as 'Pathogenic', with a given certainty, using the clinical description of such genomic variant present in the OMIM. This tool will allow the user to narrow down the amount of clinical information collected from public databases and thus enable relevant genomic variant data selection. To develop this tool six main steps were performed:

Step 1: Dataset Construction: information retrieval and type of input: We started by constructing a dataset with clinical unstructured texts from OMIM on genomic variants with a defined classification in ClinVar. This dataset was the original input that allowed for the construction and validation of the described tool. This was a step of information retrieval, and was performed mostly using a previously developed tool in the research group, *Annotator;*

Step 2: Clinical Unstructured Text Pre-processing: the collected input from step 1, *i.e.* the information present in our original dataset, was adequately pre-processed for the usage of *Text Mining* tools;

Step 3: Definition of the dictionary of relevant biological keywords: this dictionary was based on the knowledge from the literature, where each keyword had different connotations (positive or negative) and biological implications ('Benign' or 'Pathogenic'), that were in turn translated into a numeric score;

Step 4: Term Frequency-Inverse Document Frequency (TF-IDF): we next calculated the TF-IDF, to understand the importance that each keyword has in a document. In particular, we aimed at finding new relevant keywords to add to the dictionary of keywords built in the previous step (step 3), by calculating the TF-IDF for each word present in the clinical unstructured texts collected from OMIM;

Step 5: Sentiment Analysis: this step consisted of the analysis of the sentiment expressed throughout clinical unstructured texts, using the technique 'Sentiment Analysis', to understand whether the connotation of each keyword in the dictionary of keywords was well defined in the

previous steps (step 3 and 4), *i.e.* whether the score of each keyword was adequate;

Step 6: *Machine Learning* Approaches: finally, we fine-tuned our keyword scoring strategy (steps 3-5) using *Machine Learning* approaches (Decision Trees and Random Forest approaches), for maximum accuracy of the classification of clinical unstructured texts. This step was constituted by eight sub-steps: first, second and third steps were the preparation and selection of the number of genomic variants to be part of the input dataset in the ML approach and the division into training and test dataset, that were used in the ML further steps. Fourth to the seventh steps were associated with the construction of the models based on the training dataset and the evaluation of each model built. The final step was the evaluation of the performance of the final Random Forest model built using a novel dataset of genomic variants, that which were not part of the original dataset used for model creation.

Each of these steps will be fully detailed in the next sections of this Thesis.

Step 1. Dataset Construction: information retrieval and type of input

Information retrieval was the first step of this Thesis and it was done using a tool previously developed in the group that performs genomic variants annotation, classification and interpretation – Annotator. This tool allows extracting information from clinical databases such as OMIM, ClinVar, UniProt, dbSNP, etc. This information was used as input for the initial *Text Mining* steps. For the scope of this Thesis, the input collected was a large set of clinical unstructured texts on several previously identified genomic variants from the public available database OMIM and the corresponding classification performed and available in the database ClinVar.

For the purpose of this Thesis, we have selected a large set of genomic variants classified in the database ClinVar with the following clinical interpretation terms: ‘pathogenic’, ‘likely pathogenic’, ‘likely benign’, ‘benign’, ‘drug response’, ‘risk factor’, ‘protective’, ‘sensitivity’, ‘affects’ and ‘association’. These clinical interpretations provided by ClinVar are based on the American College of Medical

Genetics guidelines [66] or defined by ClinVar itself. Of notice, genomic variants classified as 'Variants of Unknown Significance' (VUS) by ClinVar were not selected to be part of our original dataset, as this classification does not define the true clinical relevance of the variant.

All collected genomic variants (except VUS) were next screened in the database OMIM. Only genomic variants with a defined classification in ClinVar and with a corresponding clinical unstructured text in OMIM were selected to be part of the original dataset. The clinical unstructured texts collected from OMIM contained a detailed description derived from research articles or manual curation from OMIM curators. This is an example of a text retrieved from OMIM database:

> **"In tumor tissue of gastric cancer (see 613659), Horii et al. (1992) identified a g-to-t somatic transition in the APC gene, Chen et al. (2011) has shown that this variant results in a gly1120-glu substitution (g1120e)."**

This example of a clinical unstructured text contains information about the relationship between a genomic variant that was detected in the gene APC and the resulting disease phenotype. Several levels of information can be observed in this example:

- The information within parentheses that refers to a number '(see 613659)' corresponds to the phenotype MIM number. This number corresponds to a given disease/phenotype. For the genomic variant in the example, the disease/phenotype associated to the MIM number is 'gastric cancer intestinal included';
- The information 'Horii et al. (1992)' corresponds to a citation of a research article from which information was retrieved by OMIM (a reference).
- The information '(G1120E)' corresponds to the amino acid alterations caused by this genomic variant. In this case a glycine (G) is replaced by a glutamate (E) in the position 1120 of the protein. This information is relevant to understand the type of protein variation that may be responsible to the observed alterations of disease/phenotype.

Not present in this example, however often found in OMIM description, are several types of numeric information, such as:

- Statistic-related numbers (p-value, odds-ratio, allele frequency),
- RS numbers derived from the 1000 Genomes project and dbSNP (public available databases on genomic variants);
- RCV numbers, which correspond to ClinVar accession identifiers;
- Cytogenetic locations, such as ‘p12.3’.

After input collection from ClinVar and OMIM, the original dataset used for this Thesis contained 25,266 genomic variants, as described in **Table 9**. In particular, the original dataset was constituted by 10 ‘.txt’ files, one for each ClinVar clinical interpretation term, wherein each line entailed the OMIM clinical unstructured text for a given genomic variant.

Table 9 - ClinVar clinical interpretation terms and the respective number of genomic variants in OMIM.

Clinical Interpretation Terms in ClinVar	# Variants with Clinical Unstructured Texts collected from OMIM
Pathogenic	23,997
Benign	174
Risk Factor	733
Association	100
Drug Response	49
Protective	57
Affects	152
Likely Benign	1
Likely Pathogenic	1
Sensitivity	2
#Total Number of Variants	25,266

Step 2. Clinical Unstructured Text Pre-processing

The next step for the development of our tool was the pre-processing of the clinical unstructured text collected previously. This was the most time-consuming step in entire *Text Mining* analysis underlying the development of our tool. The pre-processing step was highly relevant for this Thesis, given that several types of information are known to be present in the collected clinical unstructured texts collected that may hamper downstream interpretation. Therefore, the pre-processing step was aimed at the reduction of the noise and size of data collected. In particular, the pre-processing step enabled:

2.1. Case folding;

2.2. Removal/Replacement functions, in order to select and remove from the information of the collected text known to be present in OMIM texts or in any unstructured text;

2.3. Plural removal;

2.4. Tokenization.

Of notice, the pre-processing step was customized to fit the content of OMIM clinical unstructured texts using both classical and non-classical *Text Mining* approaches, as described below.

Step 2.1 Case Folding

This step consisted on the conversion of all words in the clinical unstructured text into lowercase to avoid considering the same word as different. For example:

Original Clinical Unstructured Text:

“In tumor tissue of gastric cancer (see 613659), Horii et al. (1992) identified a g-to-t somatic transition in the APC gene, Chen et al. (2011) has shown that this variant results in a gly1120-glu substitution (g1120e).”

Clinical Unstructured Text after case folding:

“in tumor tissue of gastric cancer (see 613659), horii et al. (1992) identified a g-to-t somatic transition in the apc gene, chen et al. (2011) has shown that this variant results in a gly1120-glu substitution (g1120e).”

Step 2.2 Removal/Replacement functions

This step was based on several functions, that allow the removal and/or replacement of terms, expressions and punctuation that may complicate the extraction of valuable information from the clinical unstructured text. Three main strategies were selected:

- the first strategy was aimed at specific words, such as gene names;
- the second strategy was aimed at OMIM-derived information, such as research article references;
- the third strategy was aimed at removing punctuation such as, commas, semicolon, apostrophes and decimal points.

For the first strategy, we collected two lists of words/terms which were selected for removal from the clinical unstructured texts. In particular, the lists were:

1. Gene List: a list of all OMIM-derived gene names, collected from OMIM database;

2. Stop Words List: a list of words without biological relevance, such as ‘in’, ‘the’, ‘with’. This list was compiled using a list from the Python module ‘NLTK’ [89] and further curated manually.

To remove the words within the Gene List and the Stop Words List we have used the membership testing method. In this method, we use the Python 'in' operator, which checks if an element is in a list. Therefore, the 'not in' operator tests the opposite, i.e. if an element is not in the list. Of notice, this type of method was highly time-consuming. For example:

Clinical Unstructured Text after case folding:

“in tumor tissue of gastric cancer (see 613659), horii et al. (1992) identified a g-to-somatic transition in the apc gene, chen et al. (2011) has shown that this variant results in a gly1120-glu substitution (g1120e).”

Clinical Unstructured Text after case folding and first strategy for removal/replacement of specific words/terms:

“tumor tissue gastric cancer (see 613659), horii et al. (1992) identified g-to-somatic transition gene, chen et al. (2011) has shown variant results gly1120-glu substitution (g1120e).”

For the second strategy, aimed at OMIM-related information, we have used regular expressions built to identify and remove the following information:

- Bibliographical references;
- Gene names;
- MIM numbers;
- Amino acids alterations;
- Numeric information.

By definition, regular expressions are a sequence of characters that together build a search pattern. When the match between the search pattern and the text is true, the searched string is found, and further action may be defined. To increase search efficiency, the regular expression needs to be generalist enough to be able to detect as many strings as possible, but it should not be too generalist in order to find the strings that have no interest. The main reasons to use regular expressions were the versatility and the efficient search of the pattern in the text. Of notice, for this Thesis, all OMIM-related information was removed from the clinical unstructured texts and stored in parallel ‘.txt’ files, available for downstream analysis.

To replace OMIM bibliographical references, we searched for the expression ‘et al.’ followed by a parenthesis with four digits inside of it. As presented in the text example below, this motif search would be sufficient to pinpoint the reference ‘Horii et al. (1992)’. Nevertheless, alterations to this motif were also included in our tool as not always is the term “et al.” present in the unstructured text. For example, OMIM bibliographical references might be inserted within curly brackets, might include more than one author or more than one year. When a reference was found it was replaced by a numeric code. Both the original reference and the numeric code were saved in a parallel ‘.txt’ file available for downstream purposes. For example:

> *Clinical Unstructured Text after case folding and first strategy for removal/replacement of specific words:*
>
> **“tumor tissue gastric cancer (see 613659), horii et al. (1992) identified g-to-t somatic transition gene, chen et al. (2011) has shown variant results gly1120-glu substitution (g1120e).”**
>
> *Clinical Unstructured Text after case folding, first strategy for removal/replacement of specific words and second strategy for removal and replacement of bibliographical references:*
>
> **“tumor tissue gastric cancer (see 613659), Ref.1 identified g-to-t somatic transition gene, Ref.2 has shown variant results gly1120-glu substitution (g1120e).”**

For the third strategy and following the classical *Text Mining* (TM) approach, we have removed all punctuation elements such as commas, semicolons, hyphens, apostrophes and decimal points. For example:

Clinical Unstructured Text after case folding, first strategy for removal/replacement of specific words and second strategy for removal and replacement of bibliographical references:

“tumor tissue gastric cancer (see 613659), Ref.1 identified g-to-t somatic transition gene, Ref.2 has shown variant results gly1120-glu substitution (g1120e).”

Clinical Unstructured Text after case folding, first strategy for removal/replacement of specific words and second strategy for removal and replacement of bibliographical reference, third strategy removal unwanted punctuation:

“tumor tissue gastric cancer (see 613659) Ref.1 identified g-to-somatic transition gene Ref.2 has shown variant results gly1120 glu substitution (g1120e).”

Of notice, classical TM approaches also remove parentheses and endpoints. However, in OMIM-derived clinical unstructured texts the information between parentheses (as previously described) and the endpoints were not removed to avoid loss of context. To handle with the parentheses, we separated the clinical unstructured texts into two units. The first unit, named ‘larger context unit’ includes all the text that was outside the parentheses and the second unit, the ‘smaller context unit’, includes the text that was within parentheses. The previous analyses were applied to the context units separately, however, when we made the separation, the position of the ‘smaller context unit’ into the text was lost as well as the context. Therefore, we decided not to separate the clinical unstructured text into the above-mentioned context units. Endpoints lead to the same context problem. However, in this case, endpoints were crucial to perceive in the OMIM-derived clinical unstructured texts where a sentence ends. Unlike classical TM approaches and to avoid this problem, it was decided to maintain the endpoint in the further analyses, however all other unnecessary punctuation was removed.

Step 2.3 Singularization

Words have different morphological variations, such as suffixes for making a plural word [4]. Therefore, the main purpose of the singularization step was to simplify the clinical unstructured texts, minimizing the number of words. For that, we used the Python *inflection* module that enables the singularization step [90]. This module detects the termination of a word and depending if was regular or irregular, the output of the function was the singular form of such word, *e.g.* the word 'tumours' (plural) after singularization was transformed into the singular form 'tumor' because it has a regular plural. In the case of the word 'children', this entails an irregular plural and the corresponding singular form is 'child'.

This type of transformation in OMIM texts entails problems mainly because of the use of clinical expressions, *e.g.* clinical expressions that do not have the singular form or the singular is not included in Python inflection module. Another problem occurs once the Python *inflection* module [90] considers the 's' as a plural termination, so when any word ends in 's', regardless of whether it is a plural form or not, the 's' is always removed. For example:

'is' → 'i';

'homozygous' → 'homozygou'.

Therefore, sometimes the singularization implies a complete alteration in the word compromising the efficient search for a word in the text. This was the main reason why the singularization step was not used in the pre-processing step in this Thesis.

Step 2.4 Tokenization

Tokenization consists, as previous mentioned, the splitting of a text by a specific character, and is crucial to convert the data into a format that is easier for the computer to interpret [91]. For this Thesis, the character chosen to separate the text were whitespaces. The tokenization function splits the content of the input text by whitespaces and returns a list where

each item in the list was a word or term from the input text that has been split and was named token. For example:

> *Clinical Unstructured Text after case folding, first strategy for removal/replacement of specific words and second strategy for removal and replacement of bibliographical reference, third strategy removal unwanted punctuation:*
>
> **"tumor tissue gastric cancer (see 613659), Ref.1 identified g-to-t somatic transition gene, Ref.2 has shown variant results gly1120-glu substitution (g1120e)."**
>
> *Clinical Unstructured Text after case folding, first strategy for removal/replacement of specific words and second strategy for removal and replacement of bibliographical reference, third strategy removal unwanted punctuation, tokenization:*
>
> **" 'tumor', 'tissue', 'gastric', 'cancer', '(see 613659)', 'Ref.1', 'identified', 'g-to-t', 'somatic', 'transition', 'gene', 'Ref.2', 'has', 'shown', 'variant', 'results', 'gly1120', 'glu', 'substitution', '(g1120e)', '.' "**

Unstructured clinical texts that were previously 'strings', after tokenization become a 'list of strings'. Therefore, after tokenization, the 10 '.txt' files created as described in Step 1 ('Dataset Construction: information retrieval and type of input') of this Chapter, in each line was a string (the clinical unstructured text derived from OMIM), now become a 'list of strings', *i.e.* a list of all the words that were previously separated by whitespaces. This type of transformation improves the efficiency of the downstream analysis.

In summary, for the pre-processing step of our tool, we performed a customized clean-up of the clinical unstructured texts from our original dataset, using some but now all classical techniques.

Step 3: Definition of the dictionary of relevant biological keywords

To analyse the clinical unstructured texts from OMIM, now constituted by a list of words obtained after the pre-processing described in the previous section, we decided to implement a strategy that is not classically associated with TM [70]. In particular, we opted for a strategy focused in the search for specific words in the '.txt' files generated for each type of ClinVar classification. This simpler strategy derived from our prior knowledge that certain keywords have different connotations and biological implications that can directly help in the understanding of the meaning of a given clinical unstructured text. For example, if within a clinical unstructured text from a given genomic variant the keyword 'polymorphism' occurs, this variant is likely to classified as 'Benign' by the database ClinVar classification. This classification was given once the keyword 'polymorphism' is known to have a positive connotation, *i.e.* not associated with disease. The opposite is valid for the keyword 'mutation': if it occurs within the clinical unstructured text from a given genomic variant, it is likely to be classified as 'Pathogenic' by the database ClinVar classification, as it is likely to be associated with a disease.

Other keywords exist to which it is more complicated to associate a positive or negative connotation: for example, the keyword 'substitution' cannot be clearly associated or not with disease, hence we considered it as entailing a neutral connotation. Other examples can be found in **Table 10**.

Table 10 – Connotation associated with the keywords.

Connotation	**Positive**	**Negative**	**Neutral**
Examples of Keywords	'polymorphism '	'mutation'	'homozygotes'
	'benign'	'autosomal-recessive'	'substitution'
	'natural-variant'	'susceptibilities'	'missense'

With this positive/negative/neutral connotation concept, we next decided to create a dictionary of relevant biological keywords. This initial dictionary was based on knowledge extracted from literature and careful examination of the clinical unstructured texts extracted and was constituted by 127 keywords plus

the plural of some of them, totalling 254 keywords. To understand whether other keywords could be relevant for our dictionary of relevant biological keywords, we next performed a series of measurements of the frequency of N-grams in the pre-processed clinical unstructured texts. N-grams are widely used in TM analysis [75] and by definition, are combinations of one or more words, with the three most common levels being:

- Monogram = 1 word, *e.g.* 'risk';
- Bi-gram = 2 words, *e.g.* 'increased risk';
- Tri-gram = 3 words, *e.g.* 'associated increased risk'.

Pinpointing the monogram word 'risk' or the bi-gram words 'increased risk' have distinct biological meanings, hence the relevance of analysing the existence of N-grams. Therefore, we analysed the frequency of N-grams as the main measurement that will determine whether a given N-gram will be added or not to our dictionary of relevant biological keywords. For example, if a given bi-gram was found in very low frequency in the clinical unstructured texts, adding it to the dictionary of relevant biological keywords may compromise the efficiency of our search for meaning, without adding information that another keyword could already reveal. For example, having both the keyword monogram 'susceptibility' and the bi-gram 'decreased susceptibility' in our dictionary was relevant, once both N-grams add relevant information. However, the tri-gram 'very decreased susceptibility' adds complexity to the search without adding biological meaning. Moreover, even if a given N-gram was found in high frequency, its addition to the dictionary may not bring extra relevant information. Hence, our dictionary of relevant biological keywords must be generalist enough to allow a fast understanding of the information within the clinical unstructured texts and prevent over-analysis of such texts which could compromise its efficiency. In addition, the analysis of the frequency of N-grams was also important to understand whether a given word/words have a positive or negative connotation. In particular, we generally searched for:

- high frequency N-grams in the clinical unstructured texts from variants classified as 'Benign', to which a positive connotation could be attributed and thus added to our dictionary;

- high frequency N-grams in the clinical unstructured texts from variants classified as ‘Pathogenic’, to which a negative connotation could be attributed and thus added to our dictionary.

Therefore, the frequency of N-grams was analysed to determine which monogram, bi-grams and tri-grams were most frequent in the two ‘.txt’ files of pre-processed clinical unstructured clinical texts for variants classified as ‘Benign’ (**Figure 6**) or as ‘Pathogenic’ (**Figure 7**).

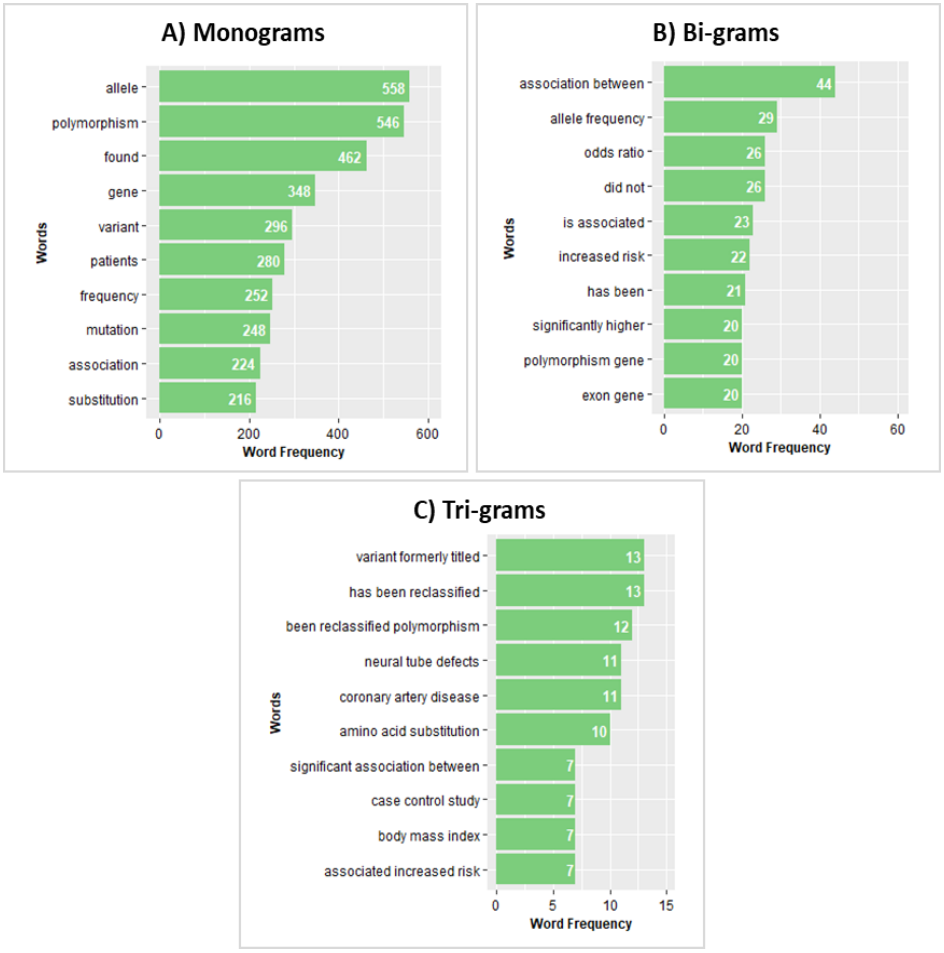


Figure 6 - Top 10 of the most frequent monograms (**A**), bi-grams (**B**) and tri-grams (**C**) in ‘Benign’ ClinVar classification.

In **Figure 6A** the monograms with the highest frequency in ‘Benign’ variants were ‘allele’ and ‘polymorphism’. These monograms are directly related, respectively, with the bi-grams ‘allele frequency’ and ‘polymorphism gene’, which are on the top 10 of the most frequent bi-grams in the pre-processed clinical unstructured texts from variants classified as ‘Benign’ (**Figure 6B**). Furthermore, the word

‘polymorphism’ is also indirectly related with the bi-gram ‘allele frequency’: in fact, when an OMIM text mentions a high frequency of alleles, this often refers to a polymorphism, as this type of genomic variants have a high frequency in the population without any malignant effects. Therefore, because of its high frequency in the pre-processed clinical unstructured texts of variants classified as ‘Benign’, the word ‘polymorphism’ was a good candidate to be added to the dictionary of relevant biological keywords with a positive connotation. Interestingly, this keyword was already part of the dictionary of relevant biological keywords, derived from knowledge extracted from literature.

Other words represented in **Figure 6A**, such as ‘found’, ‘gene’, ‘variant’, ‘patients’, ‘frequency’ were not considered as relevant keywords for our dictionary with a positive connotation because they are very common in both ‘Benign’ and ‘Pathogenic’ analysed texts and may lead to loss of search specificity (**Figure 7A**). The words ‘mutation’, ‘association’ and ‘substitution’, although with high frequency (**Figure 6A**), as we take into account the knowledge of the literature, are terms that were not directly related with a ‘Benign’ classification and thus not added to the dictionary with a positive connotation. Likely these words appear in the ‘Benign’-classified analysed texts with a negative context, for example ‘not found mutation’.

The most frequent bi-gram in **Figure 6B** was the ‘odds ratio’. Therefore, 'odds ratio', it is a statistical measure that allows calculating the change of a certain individual to have or not a disease [92]. Due to this dichotomy, we opted to not add this bi-gram to our dictionary. The analysis of tri-grams in **Figure 6B** did not introduce any relevant words to be added to our dictionary of relevant biological keywords with a positive connotation.

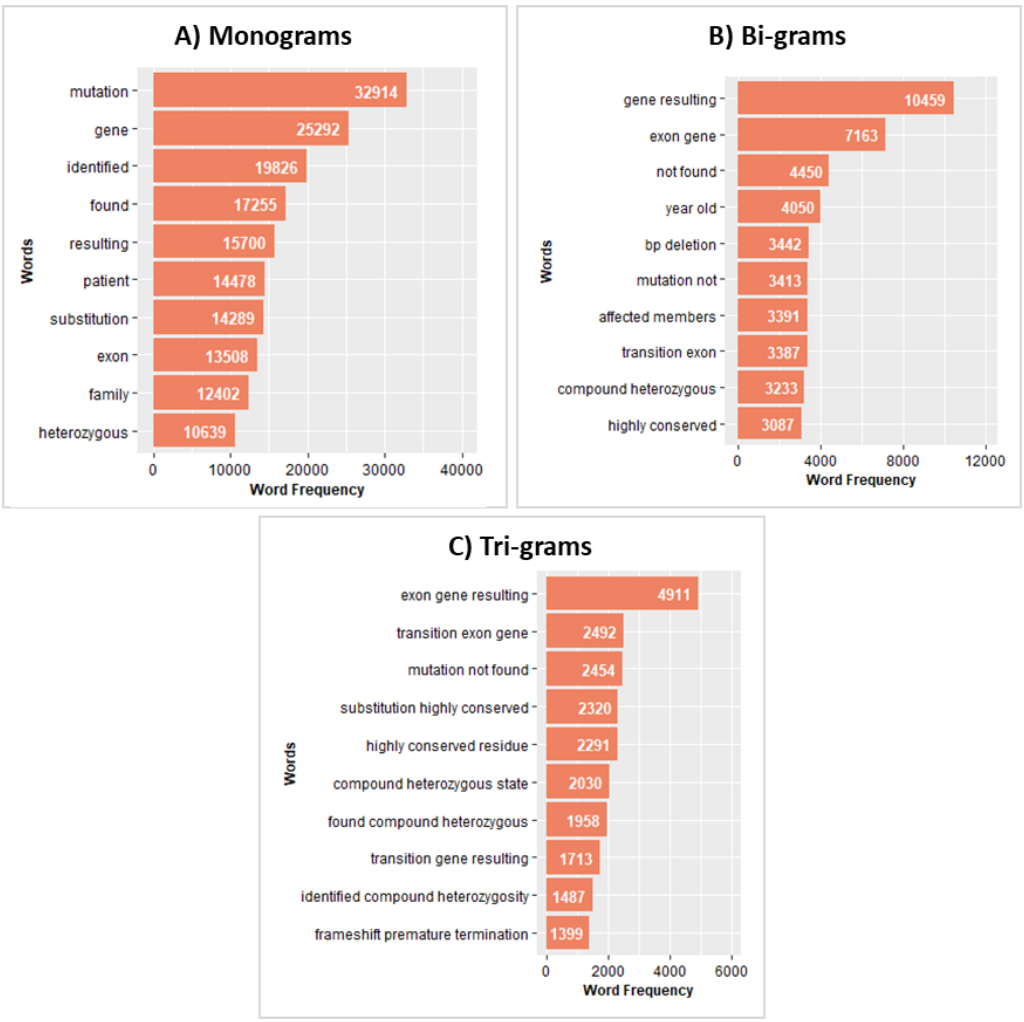


Figure 7 - Top 10 of the most frequent monograms (**A**), bi-grams (**B**) and tri-grams (**C**) in 'Pathogenic' ClinVar classification.

In the pre-processed clinical unstructured texts from genomic variants classified as 'Pathogenic', the word 'mutation' had the highest frequency (**Figure 7A**). In fact, if the word 'mutation' is mentioned in an OMIM text for a given genomic variant, often it has a malignant effect. Therefore, this word would be a relevant keyword to be added to the dictionary with a negative connotation. Nevertheless, this keyword was already part of the dictionary of relevant biological keywords, derived from knowledge extracted from literature, as well as for the observed keyword 'polymorphism'. The word 'heterozygous' was observed in the **Figure 7A** as another of the most frequent monograms in the pre-processed clinical unstructured texts from genomic variants classified as 'Pathogenic'. Similarly, to the keywords 'polymorphism' and 'mutation', the monogram 'heterozygous' was already added to the dictionary of relevant biological keywords. However, as the

keyword ‘heterozygous’ could not be associated with either a positive or negative connotation, hence we opted to add it with a neutral connotation. Of notice, neutral-connotated keywords could become positive/negative throughout the optimization of our tool.

As observed in **Figure 7A**, (and **Figure 6A**) there were monograms, such as ‘gene’, ‘identified’, ‘exon’, ‘found’ among others, that do not have a biological meaning and therefore were not added to the dictionary of relevant biological keywords. In the **Figure 7B**, ‘compound heterozygous’ was part of the top 10 of the most frequent bi-grams in the pre-processed clinical unstructured texts from genomic variants classified as ‘Pathogenic’. This bi-gram is mentioned in the OMIM texts when two different genomic variants occur in two different alleles in the same gene. These types of genomic variants cause alterations in the protein that can lead to a disease in an individual. This bi-gram was a good candidate to be added to the dictionary of relevant biological keywords with a negative connotation. However, it was already part of the dictionary of relevant biological keywords derived from knowledge extracted from literature. The tri-grams in **Figure 7C** did not have relevant keywords for our dictionary with negative connotation.

As a strategy validation, we also performed the previous analysis for the pre-processed clinical unstructured texts from genomic variants classified as ‘Drug Response’ (**Figure 8**). This was done to understand the biological relevance of mono/bi/tri-grams in this particular type of variants that are often related with the metabolic response to a drug and therefore entail several metabolism-associated words in the corresponding clinical texts.

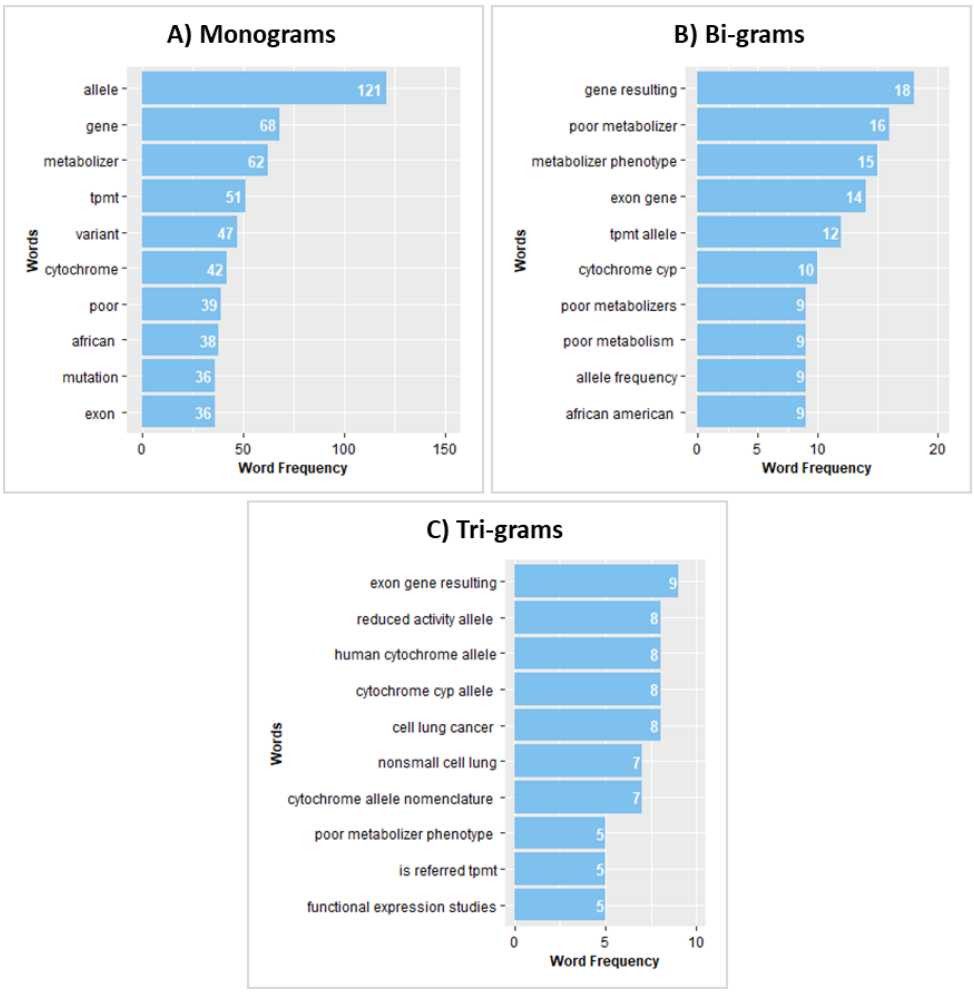


Figure 8 - Top 10 of the most frequent monograms (**A**), bi-grams (**B**) and tri-grams (**C**) in 'Drug Response' ClinVar classification.

In fact, the top-ranking words for the N-gram analysis were 'metabolizer', 'tpmt', 'cytochrome', 'poor metabolizer' or 'metabolizer phenotype' which were specific to 'Drug Response' clinical unstructured texts, *i.e.* never observed in the 'Benign' or 'Pathogenic' texts. In the case of the monogram 'metabolizer' (**Figure 8A**) and the bi-grams 'poor metabolizer', 'poor metabolizers' and 'poor metabolism' (**Figure 8B**) the high frequency is related with the four categories for the metabolism capacity: poor metabolizer, intermediate metabolizer, normal metabolizer or rapid and ultra-rapid metabolizer. If an individual is a poor metabolizer for a certain drug, he/she will need a lower dose compared to an ultra-rapid metabolizer [93]. The monogram 'tpmt' which also appears as bi-gram 'tpmt allele', is also specific for 'Drug Response' genomic variants and is the abbreviation for the enzyme thiopurine methyltransferase. This is an enzyme

responsible for the inactivation of a class of drugs named thiopurines, commonly used in inflammatory diseases. An individual with a genomic variant that lead to TPMT deficiency accumulate toxic levels of the drug since the inactive form of TPMT is not able to metabolize the drug and can be fatal [94]. The word 'cytochrome' appears in the context of the genetic tests that are performed to understand how an individual will respond to a given drug [95]. The 'cytochrome' has a high frequency in the **Figure 8A**, because cytochrome P450 is a family of enzymes responsible for the metabolism of a wide range of drugs. Furthermore, the bi-gram 'cytochrome cyp' also appears, as 'cyp' is an abbreviation of cytochrome P450 [96].

Also interesting was the monogram 'african' and the bi-gram 'african american' which appeared with a high frequency in the top-ranking of 'Drug Response' variants. This occurs because of the genetic alterations in the African population over thousands of years which may have compromised the ability of the cytochrome P450 enzyme to metabolize certain drugs. The fact that most drugs are developed based on genetic profiles of Asian and Caucasian populations may have made the drugs ineffective or increase the risk of toxicity to the African population [93],[97]. Of notice, the monogram 'allele' was transversal to all the three classifications studied ('Benign', 'Pathogenic' and 'Drug Response') as shown in **Figures 6**, **7** and **8**. Altogether, the N-grams analysis allowed:

- to verify if the keywords that were selected based on the knowledge of the literature had a high frequency in the text and thus, validate their importance;
- to find new and relevant keywords that could be added to the dictionary.

Importantly, with the N-grams analysis, no new keywords were added, because all relevant keywords were already part of the dictionary built based on literature knowledge.

Step 4: Term Frequency-Inverse Document Frequency (TF-IDF):

The TF-IDF was calculated with the *bind_tf_idf* function from the R package *tidytext*, where the input was the *one-token-per-row*, per document. **Figure 9** represents the output for this function:

```
# A tibble: 65,769 x 7
  clinVar_class word              n   total      tf   idf  tf_idf
  <chr>         <chr>         <int>   <int>   <dbl> <dbl>   <dbl>
1 Drug Response tpmt             51    4449 0.0115  1.10  0.0126
2 Drug Response cyp2d6           20    4449 0.00450 1.10  0.00494
3 Drug Response cyp2c19          14    4449 0.00315 1.10  0.00346
4 Drug Response metabolizer      11    4449 0.00247 1.10  0.00272
```

Figure 9 - Example of the output of *bind_tf_idf* function

As an example, **Table 11** and **Table 12** constitute a representation of the word counts from two different documents from our original dataset, *i.e.* the number of times each given word appears in all the pre-processed clinical unstructured texts for the genomic variants classified as 'Benign' or as 'Pathogenic'.

- the first column (ClinVar_class) contains the name of the documents, *i.e.* the ClinVar classification of the selected genomic variants;

- second column (word) contains the tokens/words, *i.e.* each of the words in the pre-processed unstructured clinical texts;

- third column (n) contains the number of times a given token/word appears in the documents, *i.e.* the number of times each word appears in the pre-processed unstructured clinical texts;

- fourth column (total) contains the number of token/words in a given document;

- the fifth, sixth and seventh columns correspond to the TF, IDF and TF-IDF measures, respectively.

Table 11 - Word counts in ‘Benign’ document.

Benign	# Total words
‘polymorphism’	273
‘consanguineous’	0
TOTAL	18,640

Table 12 - Word counts in ‘Pathogenic’ document.

Pathogenic	# Total words
‘polymorphism’	402
‘consanguineous’	3067
TOTAL	1333,874

For the word 'polymorphism' the TF-IDF calculation was made as follows:

$$\boldsymbol{TF('polymorphism', Benign) = \frac{273}{1860} \approx 0.15}$$

$$\boldsymbol{TF('polymorphism', Pathogenic) = \frac{402}{1333874} \approx 0.0003}$$

In terms of frequency, TF is the frequency of the word 'polymorphism' in each document considering the total number of words in the document. In particular, the word 'polymorphism' appears 273 times in the clinical unstructured texts from ‘Benign’ variants and 402 times in the unstructured texts from ‘Pathogenic’. However, the TF in ‘Pathogenic’ unstructured texts was lower than in ‘Benign’ unstructured texts, because the total of words in the former was larger.

For a given word, the IDF is constant for the corpus, *i.e.* for the unstructured texts from ‘Benign’ and from 'Pathogenic' variants. By definition, the IDF is the proportion of documents that present (at least once) a given word, such as 'polymorphism'. In our example, for the word ‘polymorphism’, the corpus was equal to two as both the clinical unstructured texts from ‘Benign’ and from 'Pathogenic' variants display at least once the word ‘polymorphism’.

$$\boldsymbol{IDF('polymorphism', Corpus) = log_{10}\frac{2}{2} = 0}$$

For the calculation of the TF-IDF for the word ‘polymorphism’, the product of the TF and of the IDF values in each document is performed as shown below:

$$\boldsymbol{TF - IDF('polymorphism', 'Benign') = 0.15 \times 0 = 0}$$

$$\boldsymbol{TF - IDF('polymorphism', 'Pathogenic') = 0.00030 \times 0 = 0}$$

Concerning the word 'polymorphism' it had a TF-IDF of zero for both documents. This is because the TF-IDF is a measure of the importance of words in a set of documents, *i.e.* it takes into account the frequency of a given word in two or more documents. Words that have a TF-IDF of zero are common and considered not informative. Recalling the monogram analysis, the word ‘polymorphism’ was considered relevant and with a positive connotation, as it ranked in the top10 of frequencies in the clinical unstructured texts from ‘Benign’ variants (**Figure 6A**) and was absent from the top 10 of frequencies in the clinical unstructured texts from ‘Pathogenic’ variants (**Figure 7A**). However, the TF-IDF analysis showed that the word ‘polymorphism’ was in fact common in both types of variants (TF-IDF = 0 for both). This shows the relevance of the TF-IDF strategy overall and the limitations of the N-gram analysis. Nevertheless, we opted not to remove this word as it could be present in the clinical unstructured texts from ‘Pathogenic’ variants with a negative context, for example ‘not a polymorphism’.

Another example for the TF-IDF analysis was the word 'consanguineous', which occurred 3067 times in the ‘Pathogenic’ unstructured texts and was absent from the ‘Benign’ unstructured texts. Therefore, the TF calculation was:

$$TF('consanguineous','Benign') = \frac{0}{1860} = 0$$

$$TF('consanguineous','Pathogenic') = \frac{3067}{1333874} \approx 0.002$$

As expected, the TF in the ‘Benign’ unstructured texts was zero, because the word ‘consanguineous’ was never detected. The calculation of IDF for this word took into account that fact, *i.e.* that the word ‘consanguineous’ only occurred in one of the two documents in the corpus.

$$IDF('consanguineous', Corpus) = log_{10}\frac{2}{1} = 0.30$$

Therefore, the TF-IDF calculation was:

$$TF - IDF('consanguineous','Benign') = 0 \times 0.30 = 0$$

$$TF - IDF('consanguineous','Pathogenic') = 0.002 \times 0.30 = 0.001$$

Hence, the word 'consanguineous' was not a frequent word in the clinical unstructured texts of genomic variants classified as 'Pathogenic'. Moreover, this word did not appear in the top 10 of the most frequent monograms (**Figure 7A**). Consequently, the TF-IDF of this word showed its particular relevance with the clinical unstructured texts of genomic variants classified as 'Pathogenic' and important to be added to our dictionary of relevant biological keywords.

The TF-IDF technique was used to analyse the importance of all words in the clinical unstructured texts of genomic variants classified as ‘Benign’ or as 'Pathogenic' (**Figure 10A, B**), to try to find new keywords to add to the dictionary of relevant biological keywords. As with the N-gram analysis, we also analysed the unstructured texts for genomic variants classified as ‘Drug Response’, as a

validation strategy (**Figure 10C**). Of notice, this type of analysis did not take into account the biological context and only focused in numerical statistic. Another important observation was that many rounds of text clean-up were performed to remove misspelled sets of letters with high TF-IDF that derived from the pre-processing steps described previously, and without biological meaning for a TM strategy, such as '-95delc' and '2240del12'.

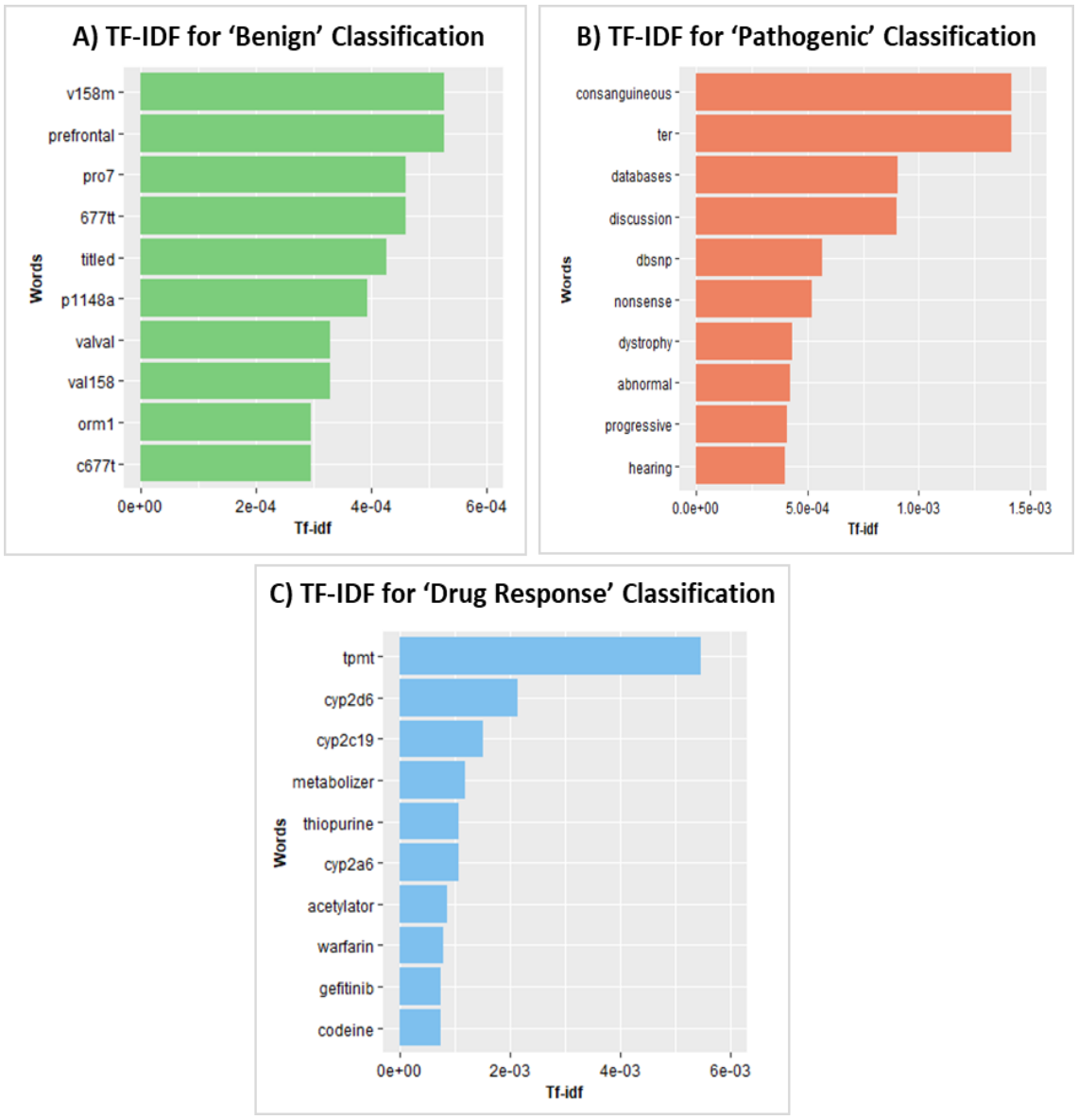


Figure 10 - Term Frequency-Inverse Document Frequency (TF-IDF) for the 'Benign' (**A**), 'Pathogenic' (**B**) and 'Drug Response' (**C**) ClinVar classification.

As visible in **Figure 10**, there was no word in the bar plots that could be considered relevant: for example, the top word ‘schizophrenia’ (for the ‘Benign’ clinical unstructured texts) could not be included in our dictionary as it was too specific and did not entail any relevant biological meaning (**Figure 10A**). A similar rationale could be made for all other top-ranking words for both the ‘Benign’ and ‘Pathogenic’ clinical unstructured texts (**Figure 10B**). In addition, several ‘nonsense’ words remained, even after the multiple rounds of text clean-up such as ‘lh’ and ‘ter’, which refers to a hormone (Luteinizing hormone, **Figure 10A**) or to a protein (Ter protein, **Figure 10B**), respectively. Like the word ‘schizophrenia’, ‘lh’ and ‘ter’ were too specific to belong to our dictionary. Hence, no novel word was added to our dictionary with this TF-IDF approach. Nevertheless, the TF-IDF analysis for the ‘Drug Response’ unstructured texts reproduced and thus validated our observations in the N-gram analysis (**Figure 10C**) for example, the high ranking observed for the enzymes ‘tpmt’, ‘cyp’ and the word ‘metabolizer’, recall the observations with the N-gram strategy.

In summary, the TF-IDF analysis did not allow the addition of any new keywords to our dictionary of relevant biological keywords. Therefore, our dictionary was constituted by the same 254 (dictionary keywords (*n*=127) plus respective negative form (*n*=127)) keywords based solely on literature knowledge.

Step 5: Sentiment Analysis

Sentiment analysis is a technique used to extract “emotions” that are expressed throughout a text. The words that appear in a given text are classified with positive or negative scores depending on the sentiment/emotion [98]. A classic example was the comparison of the first and last books from the “Harry Potter” series: the text in the first book was associated with more a positive emotion, with a high frequency of words such as ‘love’ and ‘friendship’; the text in the last book was more associated with negative emotions, with a high frequency of words such as ‘battle’ and ‘death’ [99].

Our strategy was to perform the sentiment analysis using the pre-processed clinical unstructured texts to measure the overall sentiment for the ‘Benign’ and

‘Pathogenic’-classified variants. Therefore, we expected that a genomic variant classified as ‘Benign’ would have a positive sentiment and a genomic variant classified as ‘Pathogenic’ would have a negative sentiment associated. For this, we used the same R package previously mentioned, the tidytext. This R package contains three sentiment lexicons, “AFFIN” [100], “bing” [101] and “nrc” [102] which are three list of monograms with an associated sentiment. In the case of lexicon “AFFIN” each monogram has an associated sentiment score between -5 to 5 depending on if the word has a positive or a negative sentiment. The “bing” lexicon categorized the monograms in a binary system simply into positive and negative categories. Finally, the “nrc” lexicon has several categories such as, anger, anticipation, disgust, fear, joy, sadness, surprise and trust. However, these three lexicons are constituted only by monograms completely unrelated to the biological nature of our clinical unstructured texts. Therefore, we have used our pre-defined dictionary of relevant biological keywords to constitute our own lexicon.

To define our lexicon, we started by considering the connotation of a keyword by assigning a score, *i.e.* a keyword with a positive connotation has a positive score and a keyword with a negative connotation has a negative score. However, there are other keywords that cannot be clearly associated with a positive or negative connotation, the neutral keywords, therefore, so we assign the score zero. This strategy was used to measure the main sentiment in a given genomic variant text. Therefore, we expected that a genomic variant classified as ‘Benign’ has an overall positive score and a genomic variant classified as ‘Pathogenic’ has an overall negative score. The score for each genomic variant text was defined across several rounds of fine-tuning. In total seven distinct scoring approaches were tested, to understand the one that was more accurate, *i.e.* the one that produced the minimum false-positive/negatives count. We started by defining Score v1, as follows:

- Keywords with negative connotation, such as ‘increased-risk’ and ‘mutation’ were given a value of -1 and -2, respectively;
- Keywords with positive connotation, such as ‘benign’ and ‘polymorphism’ were given a value of +1 and +2, respectively;

- Keywords with a neutral connotation, such as ‘heterozygous’, were given a value of 0;

Keyword repetition in the same clinical unstructured text was not taken into account, *e.g.* in the example text “(…) it is a polymorphism detected in the normal population. Furthermore, this polymorphism was absent from the African population (…)”, the keyword ‘polymorphism’, although appearing twice in the text, for the scoring strategy the value of +2 was only accounted once. As we performed the sentiment analysis with the Score v1, we obtained the results presented in **Table 13**:

Table 13 - Results from the sentiment analysis with the Score V1.

Variant ClinVar Classification	**Number of Variants**			
	Score > 0	**Score < 0**	**No Keywords found**	**Score = 0**
Benign (*n*=174)	69 (39.7%)	44 (25.3%)	26 (14.9%)	35 (20.1%)
Pathogenic (*n*=23,997)	113 (0.5%)	19,601 (81.7%)	408 (1.7%)	3875 (16.1%)

Considering the sentiment analysis with Score v1, we observed the following results (**Table 13**):

- For the ‘Benign’ variants:
    - 69 true-positives (TP), *i.e.* ‘Benign’ variants with positive overall score;
    - 44 false-positives (FP), *i.e.* ‘Benign’ variants with negative overall score;
- For the ‘Pathogenic’ variants:
    - 19,601 true-negatives (TN), *i.e.* ‘Pathogenic’ variants with negative score;
    - 113 false-negatives (FN), *i.e.* ‘Pathogenic’ variants with positive overall score.

However, several genomic variants presented with overall score of zero. We opted to divide these genomic variants in two categories:

- ‘No Keywords found’, this category includes the genomic variants where the clinical unstructured text does not have dictionary keywords;
- ‘Score = 0’, this category includes the genomic variants in which the sum of values given to the dictionary keywords found in the clinical unstructured text was zero.

Concerning the results of Score v1 for the variants with a ‘Benign’ classification a score of zero, we observed 26 genomic variants without any dictionary keywords in the corresponding clinical unstructured texts and 35 where the sum of values given to the dictionary keywords found in the clinical unstructured text was zero (**Table 13**). For the variants with a ‘Pathogenic’ classification, 408 did not have any keyword in the clinical unstructured texts and 3875 had a sum of values given to the dictionary keywords found in the clinical unstructured text of zero (**Table 13**).

To understand the ability of the scoring approach to correctly score a genomic variant we performed the percentage of correct scoring, that consisted of the number of TP or TN divided by the total number of genomic variants for each classification. Therefore, in the case of Score v1, the percentage of correct scoring of ‘Benign’ variants were 39.7% and for the ‘Pathogenic’ variants were 81.7% (**Table 13**).

To optimize the number of genomic variants correctly classified we next developed and tested seven distinct scoring approaches, all deriving from Score v1. Score v2 had the following criteria:

- Keywords with negative connotation, such as ‘increased-risk’ and ‘mutation’ were given a value of -1 and -2, respectively (*equal to* Score v1);
- Keywords with positive connotation, such as ‘benign’ and ‘polymorphism’ were given a value of +1 and +2, respectively (*equal to* Score v1);
- Keywords with a neutral connotation, such as ‘heterozygous’, were given a value of 0 (*equal to* Score v1);
- Keyword repetition in the same clinical unstructured text was taken into account, e.g. in the example text “(…) it is a polymorphism detected in the

normal population. Furthermore, this polymorphism was absent from the African population (…)", the keyword 'polymorphism', as it appears twice in the text, for this scoring strategy the value of +4 (+2x2) was given for this keyword. In summary, the keyword score considered the number of times it was repeated in the text (*unlike* Score v1).

The sentiment analysis with Score v2 provided better results than Score v1, in terms of true positives and variants with a score equal to 0 (**Table 14**).

Table 14 - Results from the sentiment analysis with the Score V2.

Variant ClinVar Classification	**Number of Variants**			
	Score > 0	**Score < 0**	**No Keywords found**	**Score = 0**
Benign (*n*=174)	73 (41.9%)	42 (24.1%)	26 (14.9%)	3 (19.0%)
Pathogenic (*n*=23,997)	110 (0.5%)	19,613 (81.7%)	408 (1.7%)	3866 (16.1%)

Taking into account the values observed in **Table 14**, the sentiment analysis with Score v2, for variants with a 'Benign' classification correctly classified 73 genomic variants (true-positives) and incorrectly 42 (false-positives). For the 'Pathogenic' variants Score v2 classified correctly 19,613 genomic variants (true-negatives) and incorrectly 110 (false-negatives). These results with Score v2 represented an improvement in classification in comparison with the results from Score v1. Another improvement with Score v2 (*vs.* Score v1) was in the number of genomic variants where the sum of the values of each dictionary keyword found in the clinical unstructured text was zero: 33 'Benign' classified variants presented a sum equal to 0 with Score v2 (*vs*. 35 for Score v1) and; 3866 'Pathogenic' classified variants presented a sum equal to 0 with Score v2 (*vs.* 3875 for Score v1). This showed that Score v2 was able to increase the number of true-positives/true-negatives and to diminish the number of genomic variants with a sum equal to 0 in comparison with Score v1. This improvement was due to the

fact that several genomic variants presented the same keyword more than once which, unlike Score v1, was accounted for in Score v2.

Considering the classic approaches to TM, we knew *a priori* that subtasks exist that should be taken into account during the sentiment analysis. Therefore, Score v3 included an important subtask in classical TM approaches, the 'Negation Handling'. Therefore, we proceeded to a sentiment analysis with Score v3 that had the following criteria:

- Keywords with negative connotation, such as 'increased-risk' and 'mutation' were given a value of -1 and -2, respectively (*equal to* Score v1 and v2);
- Keywords with positive connotation, such as 'benign' and 'polymorphism' were given a value of +1 and +2, respectively (*equal to* Score v1 and v2);
- Keywords with a neutral connotation, such as 'heterozygous', were given a value of 0 (*equal to* Score v1 and v2);
- Keyword repetition in the same clinical unstructured text was considered (*equal to* Score v2);
- The presence of a 'negation word' in the immediate vicinity of a keyword was considered. For example, if the clinical unstructured text contains an expression such as '(…) not found mutation (…)' or '(…) it is not polymorphic (…)', the value of the keyword was reversed, *i.e.* the value of -2 for the keyword 'mutation' became a value of +2 or the value of +2 for the keyword 'polymorphic' became a value of -2 (*unlike* Score v2);

This last criterium was extremely relevant as the frequency of a keyword in a genomic variant text was not the only factor that alters the sentiment analysis. The negation was also important as it inverts the sentiment in a text. Negation words such as 'no', 'not', 'cannot' and 'shouldn't' were examples used in the 'Negation Handling' to understand each portion of the sentence affected by the negation. This portion is named 'vicinity' or 'scope negation' [2]. Handling the negation is challenging because a negation in a simple sentence may invert the connotation of all words, however, in a compound sentence a negation usually only inverts the connotation of some words [2]. Before handling the negation, it was necessary to make alterations in the negative words in the clinical

unstructured texts. Abbreviations such as ‘didn’t’, ‘shouldn’t’, ‘aren’t’ and so forth, were transformed into the complete form, such as ‘did not’, ‘should not’ and ‘are not’. This type of transformation was important to increase the efficiency and decrease the effort and the time consumed in the analysis, because with this transformation the tool does not need to deal with multiple words with the same meaning.

To handle the impact of the negation, we defined a list of negation words which were used as an indicator of a negation in a sentence (**Table 15**). The list of negation words includes the two classes of negation words considered, *i.e.* syntactic and diminisher. The syntactic class includes all negation words that invert completely the connotation of other words, while the diminisher class includes all words that reduce the connotation rather than invert it.

Table 15 – List of Negations words [2].

Negation Class	Negations
Syntactic	no, not, rather, could not, was not, did not, would not, should not, were not, do not, does not, have not, has not, wont, had not, never, none, nobody, nothing, neither, nor, nowhere, is not, cannot, must not, might not, without, need not
Diminisher	hardly, less, little, rarely, scarcely, seldom

For the purpose of this Thesis, we opted to used only the syntactic negation class, in particular the negation words ‘no’ and ‘not’. This selection was performed after analysing a series of clinical unstructured texts from our dataset, which showed that these were the highest represented negation words. Moreover, the negation word was searched both:

- **immediately before a dictionary keyword**, for example “(…) it is not polymorphic (…)” – Score v3.1;
- **up to two words before a dictionary keyword**, for example “(…) not found mutation (…)” - Score v3.2;

Negation words were also searched farther apart from the dictionary keyword; however, no major alterations in the end result were observed. Results with Score v3 for the sentiment analysis were better than for Score v2, but still many misclassified genomic variants were observed.

Table 16 - Results from the sentiment analysis with the Score V3.

	Position of Negation Word	Variant ClinVar Classification	Number of Variants			
			Score > 0	Score < 0	No Keywords found	Score = 0
Score v3.1	***immediately before a dictionary keyword***	**Benign (*n*=174)**	73 *(41.9%)*	42 *(24.1%)*	26 *(14.9%)*	33 *(18.9%)*
		Pathogenic (*n*=23,997)	104 *(0.4%)*	19,702 *(82.1%)*	408 *(1.7%)*	3783 *(15.7%)*
Score v3.2	***up to two words before the dictionary keyword***	**Benign (*n*=174)**	73 *(41.9%)*	42 *(2.1%)*	26 *(14.9%)*	33 *(18.9%)*
		Pathogenic (*n*=23,997)	133 *(0.5%)*	19,609 *(81.7%)*	408 *(1.7%)*	3847 *(16.0%)*

In Score v3.1 and as shown in **Table 16**, genomic variants classified as 'Pathogenic' when the position of negation word was only 'immediately before a dictionary keyword' have an increase in the number of TN compared with the Score v2: 19,913 with Score v2 *vs.* 19,702 with Score v3.1. The increase in the number of TN demonstrated that the 'Negation Handling' needs to be considered in the sentiment analysis and in our scoring strategy. No alterations in the number of TP were observed at this point. As we analysed the results for negation words found 'up to two words before the dictionary keyword', we observed that we still obtained better results than with Score v2: 19,913 with Score v2 *vs.* 19,609 with Score v3.2. However, Score v3.2 revealed less genomic variants correctly scored than Score v3.1, likely due to parts of clinical unstructured texts such as the one shown in **Table 17**.

Table 17 – Example of part of an unstructured clinical text and the corresponding score for v3.1 and v3.2.

Examples of part of an unstructured clinical text	**Score v3.1**		**Score v3.2**	
	immediately before a dictionary keyword		***up to two words before the dictionary keyword***	
Example A: '(…) <u>not</u> found'	'found mutation'	1 x (-2)	'not found mutation'	1 x (+2)

Taking this example into account, the score for the keyword 'mutation' with Score v3.1 was - 2, while for Score v3.2 was +2. This was because the negation word 'not' was found in the second word before the keyword, hence only considered for Score v3.2. However, we considered that the 'up to two words before the dictionary' scoring strategy, *i.e.* Score v3.2 was likely to be more faithful: Score v3.2 was more inclusive, as it counted the occurrence of negation words immediately before and before that, unlike Score v3.1, which was blind to negation words present two words before the keyword. This was the main reason why we opted to continue to use, for the following scoring strategies, the 'up to two words before the dictionary' strategy.

Next, we defined Score v4, in which we decided to augment the range of the individual keyword scores, *i.e.* we gave a more negative/positive score to some keywords. For example:

- the keyword 'polymorphism', which was individually scored with +20 in the previous approaches, became scored with a value of '+30';
- the keyword 'benign, which was individually scored with '+10' in the previous approaches, became scored with a value of '+20';
- the keyword 'mutation', which was individually scored with '-20' in the previous approaches, became scored with a value of '-30';
- the keyword 'increased-risk', which was individually scored with '-2' in the previous approaches, became scored with a value of '-5';
- the keywords 'autosomal-dominant' and 'pathogenic', which were individually scored with '-10' in the previous approaches, became scored with a value of '-20'.

This enlarged range of our individual scoring of keywords was done to increase the positivity and negativity of keywords that, by visual inspection of the scoring matrices that underlie the sentiment analysis technique, seemed to be more relevant. This range alteration was therefore an attempt to increase the number of true-positives and true-negatives and diminish the number of false-positives and false-negatives obtained with the previous scoring strategies. Therefore, Score v4 had the following criteria:

- keywords with negative connotation, such as 'increased-risk', 'autosomal-dominant ', 'pathogenic', 'mutation' were given a value of -5, -10, -20 and -30, respectively (*unlike* Score v3);
- keywords with positive connotation, such as 'benign' and 'polymorphism' were given a value of +20 and +30, respectively (*unlike* Score v3);
- keywords with a neutral connotation, such as 'heterozygous', were given a value of 0 (*equal to* Scores v1, v2 and v3);
- keyword repetition in the same clinical unstructured text was considered (*equal to* Scores v2 and v3);
- search for negation words was made only 'up to two words before the dictionary keyword' (*equal to* Score v3.2).

Obtained results with Score v4 can be analysed in **Table 18**.

Table 18 - Results from the sentiment analysis with the Score v4.

Variant ClinVar Classification	***Number of Variants***			
	Score > 0	**Score < 0**	**No Keywords found**	**Score = 0**
Benign (*n*=174)	78 *(44.8%)*	40 *(23.0%)*	26 *(14.9%)*	30 *(17.2%)*
Pathogenic (*n*=23,997)	168 *(0.7%)*	19,629 *(82.0%)*	408 *(1.7%)*	3792 *(15.8%)*

With Score v4 the number of TP and TN increased in comparison with Score v3.2. In particular:

- the number of TP increased from 73 to 78 correctly classified 'Benign' variants;

- the number of TN increased from 19,609 to 19,629 correctly classified ‘Pathogenic’ variants.

Furthermore, the number of FP decreased from 42 to 40 incorrectly classified ‘Benign’ variants. However, the number of FN increased from 133 to 168 incorrectly classified ‘Pathogenic’ variants. Concerning genomic variants with a score of 0, Score v4 also represented an improvement from Score v3.2. In particular, both ‘Benign’ and ‘Pathogenic’ classified variants with an overall score of 0 decreased: from 33 to 30 ‘Benign’ classified variants and from 3847 to 3792 ‘Pathogenic’ classified variants (**Table 16** and **Table 18**).

The graphics of sentiment analysis allowed visualizing the differences between sentiments presented in the unstructured clinical texts. The classical sentiment analysis graphic is a 2-dimensional representation, where the x-axis represents the number of genomic variants texts classified with a given classification and the y-axis represents the overall score for each variant text. Therefore, each bar represents a genomic variant text with the overall score associated: bars below zero have an overall score negative and above zero have an overall score positive. **Figure 11** represents the obtained sentiment analysis graphics for ‘Benign’ and ‘Pathogenic’ classified variants **(Figure 11A** and **11B**, respectively).

We observed an overall positive sentiment (bars above zero) for the ‘Benign’ genomic variants and an overall negative sentiment (bars below zero) for the ‘Pathogenic’ genomic variants. Also visible in **Figure 11** were spaces without bars that represented the ‘Benign’/’Pathogenic’ genomic variants with an overall score of zero. With Score v4 we obtained the highest number of TP and TN and the minimum genomic variants with score equal to zero.

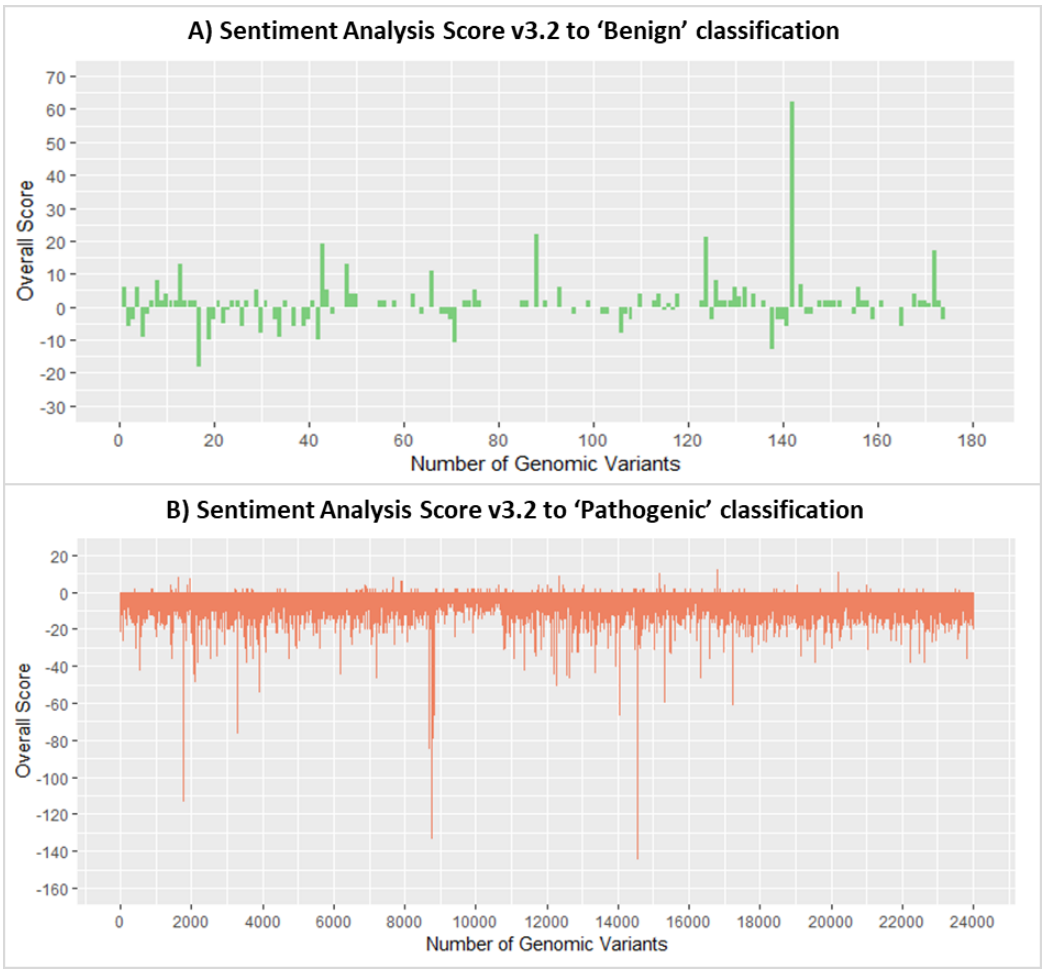


Figure 11 - Sentiment Analysis for Score v3.2 for 'Benign' (**A**) and 'Pathogenic' (**B**) ClinVar classification.

In order to further improve the number of TP and TN, we proceeded to redefine our scoring strategy, and created Score v5, which had the following criteria:

- keywords with negative connotation, such as 'increased-risk', 'autosomal-dominant ', 'pathogenic' and 'mutation' were given a value of -5, -10, -20 and -30, respectively (*equal to* Score v4);
- keywords with positive connotation, such as 'benign' and 'polymorphism' were given a value of +20 and +30, respectively (*equal to* Score v4);
- keywords with a neutral connotation, such as 'heterozygous', were given a value of 0 (*equal to* Score v4);
- keyword repetition in the same clinical unstructured text was taken into account (*equal to* Score v4);
- search for negation words was made only up to two words before the dictionary keyword (*equal to* Score v4);

- the presence of a ‘MIM number’ which corresponds to a disease/phenotype in the OMIM databases was considered in the Score v5, *e.g.* the example, “(…) in tumor tissue of gastric cancer (see 613659) and in a colorectal carcinoma (114500) (…)”, contains two different MIM numbers (613659 and 114500). However, in this scoring strategy we only considered the presence or absent of MIM numbers. Therefore, if a clinical unstructured text has at least one MIM number, the value added to the overall score was -20, independently of the number of times it may occur (*unlike* Score v4).

Obtained results with Score v5 can be analysed in **Table 19**.

Table 19 - Results from the sentiment analysis with the Score v5.

Variant ClinVar Classification	***Number of Variants***			
	Score > 0	**Score < 0**	**No Keywords found**	**Score = 0**
Benign (*n*=174)	74 *(42.5%)*	47 *(27.0%)*	26 *(14.9%)*	27 *(15.5%)*
Pathogenic (*n*=23,997)	99 *(0.4%)*	23,378 *(97.4%)*	265 *(1.1%)*	255 *(1.8%)*

In Score v5, we added a criterium that searches for the presence of a reference to a disease. In particular, a ‘MIM number’, given that most OMIM descriptions (our clinical unstructured texts) associated with disease encompass a reference to it. Therefore, if a ‘MIM number’ was found within the clinical unstructured texts, a value of -20 was added to the overall score. Of notice, the ‘MIM number’ was collected from OMIM database. With Score v5, we observed that genomic variants classified as ‘Pathogenic’ were those most altered in comparison with Score v4 (**Table 18** *vs.* **Table 19**). In particular:

- the number of TN increased from 19,629 to 23,378 genomic variants correctly classified as ‘Pathogenic’;
- the number of FN decreased from 168 to 99 genomic variants incorrectly classified as ‘Pathogenic’.

Furthermore, for the ‘Pathogenic’ variants, the number of variants with no keywords found, decreased from 408 to 265 and for the ‘Pathogenic’ variants with the score equal to zero, the number decreased from 3792 to 255 in comparison to Score v4. The alterations between results occurred because of the addition of the ‘MIM number’ criterium in Score v5.

For the variants classified as ‘Benign’ and considering the last criterium of Score v5, the number of TP decreased from 78 to 74 and the number of FP increased from 40 to 47 in comparison with Score v4. The ‘Benign’ genomic variants with score equal zero decreased from 30 to 27 and the ‘Benign’ variants with no keywords found did not show alterations (**Table 18** *vs.* **Table 19**). This showed that the ‘MIM Number’ criterium shifted the overall score for more negative values, *i.e.* towards pathogenicity.

In order to increase the number of genomic variants correctly classified, we designed Score v6, which had the following criteria:

- keywords with negative connotation, such as ‘increased-risk’, ‘autosomal-dominant ‘, ‘pathogenic’ and ‘mutation’ were given a value of -5, -10, -20 and -30, respectively (*equal to* Score v5);
- keywords with positive connotation, such as ‘benign’ and ‘polymorphism’ were given a value of +20 and +30, respectively (*equal to* Score v5);
- keywords with a neutral connotation, such as ‘heterozygous’, were given a value of 0 (*equal to* Score v5);
- keyword repetition in the same clinical unstructured text was taken into account (*equal to* Score v5);
- search for negation words was made only up to two words before the dictionary keyword (*equal to* Score v5);
- search for ‘MIM number’, however in this score (Score v6) we considered the number of times a different ‘MIM number’ occurred in a clinical unstructured text. Therefore, the value added to the overall score of a genomic variant, for example, with three different ‘MIM numbers’ was -60, because we multiplied the number of times a different ‘MIM number’ occurred in a clinical unstructured text by -20 (3 x -20) (*unlike* Score v5).

Obtained results with Score v6 can be analysed in **Table 20**.

Table 20 - Results from the sentiment analysis with the Score v6.

Variant ClinVar Classification	*Number of Variants*			
	Score > 0	**Score < 0**	**No Keywords found**	**Score = 0**
Benign (*n*=174)	73 *(42.0%)*	48 *(27.6%)*	26 *(14.9%)*	27 *(15.5%)*
Pathogenic (*n*=23,997)	97 *(0.4%)*	23,383 *(97.4%)*	265 *(1.1%)*	252 *(1.1%)*

With Score v6 we improved the results for the genomic variants classified as 'Pathogenic' in comparison with Score v5. In particular:

- the number of variants correctly classified as 'Pathogenic' increased from 23,378 to 23,383;
- the number of variants incorrectly classified as 'Pathogenic' decreased from 99 to 97.

Moreover, the results for the 'Pathogenic' variants with score equal to zero decreased from 255 to 252 and in the end the variants with no keywords found did not show an alteration, when compared with Score v5 (**Table 19** *vs.* **Table 20**). These results showed that the search for different 'MIM numbers' in clinical unstructured texts, improved the results of Score v6. For the 'Benign' genomic variants, the number of TP decreased from 74 to 73 and the number of FP increased from 47 to 48 genomic variants in comparison with Score v5. The number of 'Benign' variants with no keywords found and with a score equal to zero did not show alterations (**Table 19** *vs.* **Table 20**). Similarly, to Score v5 and Score v6 was also designed towards pathogenicity, favouring the classification of 'Pathogenic' genomic variants and disfavouring the classification of 'Benign' genomic variants. To address this, we designed Score v7, which had the following criteria:

- keywords with negative connotation, such as 'increased-risk', 'autosomal-dominant ', 'pathogenic' and 'mutation' were given a value of -5, -10, -20 and -30, respectively (*equal to* Score v6);

- keywords with positive connotation, such as ‘benign’ and ‘polymorphism’ were given a value of +20 and +30, respectively (*equal to* Score v6);
- keywords with a neutral connotation, such as ‘heterozygous’, were given a value of 0 (*equal to* Score v6);
- keyword repetition in the same clinical unstructured text was taken into account (*equal to* Score v6);
- search for negation words was made only up to two words before the dictionary keyword (*equal to* Score v6);
- search for ‘MIM number’, however, in this score (Score v6) we considered all the times that any ‘MIM number’ occurred in a clinical unstructured text. Therefore, we did not distinguish between repeated ‘MIM numbers’ and those that appeared only once, e.g. in a clinical unstructured text with four ‘MIM numbers’ where two of them were repeated the value added to the overall score was -80 (4 x -20) (*unlike* Score v6).

Obtained results with Score v7 can be analysed in **Table 21**.

Table 21 - Results from the sentiment analysis with the Score v7.

Variant ClinVar Classification	***Number of Variants***			
	Score > 0	**Score < 0**	**No Keywords found**	**Score = 0**
Benign (*n*=174)	73 *(42.0%)*	48 *(27.6%)*	26 *(14.9%)*	27 *(15.5%)*
Pathogenic (*n*=23,997)	97 *(0.4%)*	23,383 *(97.4%)*	265 *(1.1%)*	252 *(1.1%)*

By comparing **Table 20** (Score v6) and **Table 21** (Score v7), we observed that results for ‘Benign’ and ‘Pathogenic ‘genomic variants were equal to both Score v7 and v6. **Figure 12** represents the sentiment analysis graphics for ‘Benign’ and ‘Pathogenic’ classified variants for Score v7 (**Figure 12A** and **12B**, respectively). We observed in **Figure 12A** an overall positive sentiment (bars above 0) for the genomic variants classified as ‘Benign’ and in **Figure 12B** an overall negative sentiment (bars below 0) for the genomic variants classified as ‘Pathogenic’.

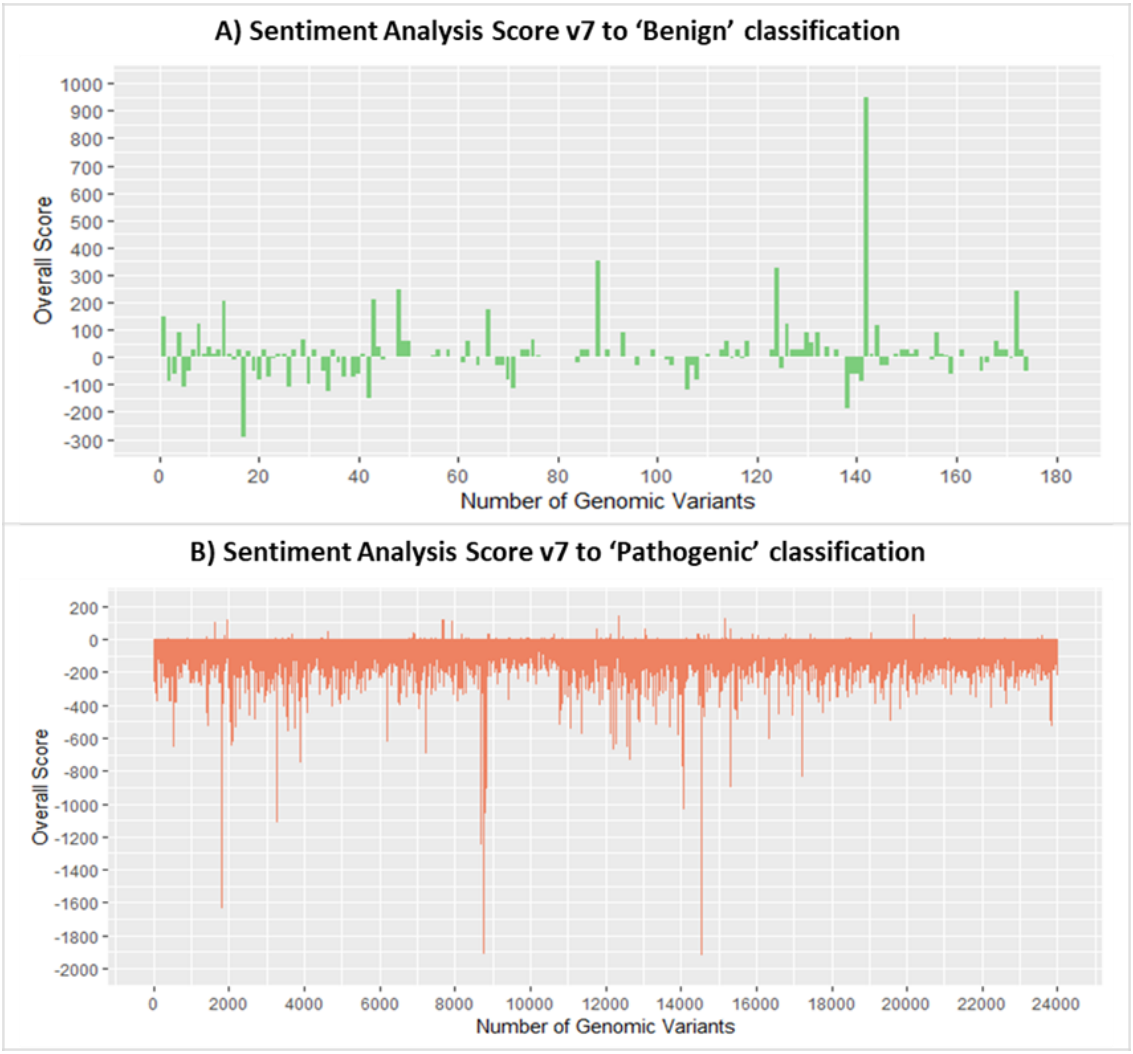


Figure 12 - Sentiment Analysis for Score v7 to 'Benign' (**A**) and 'Pathogenic' (**B**) ClinVar classification.

We expected that with the alterations in the criteria from Score v6 to Score v7, we would have a significant increase of correctly classified genomic variants. However, as observed in **Table 20** (Score v6) and **Table 21** (Score v7), there was no increase in the number of genomic variants correctly classified.

Table 22 summarizes all the results of the sentiment analysis performed for each scoring approach (Score v1-v7). For simplicity, in **Table 22** the 'Correctly Classified' column refers to the number of true positive genomic variants ('Benign' variants correctly classified, *i.e.* with a Score > 0) added to the number of true negative genomic variants ('Pathogenic' variants correctly classified, *i.e.* with a Score < 0). Furthermore, also in **Table 22**, the 'Misclassified' column refers to the

number of false positive genomic variants ('Pathogenic' variants incorrectly classified as 'Benign', *i.e.* with a Score > 0) added to the number of false negative genomic variants ('Benign' variants incorrectly classified as 'Pathogenic', *i.e.* with a Score < 0). In **Table 22**, the 'No Keyword found/Score = 0' column refers to the number of genomic variants with no keywords found in the corresponding clinical unstructured texts added to the number of genomic variants with a score equal to 0.

Table 22 – Results of the sentiment analysis performed for each scoring approach (Score v1-v7).

Scoring Approaches	Correctly classified (TP +TN)	Misclassified (FP + FN)	No Keyword / Score = 0
Score v1	19,670	157	4344
Score v2	19,686	152	4333
Score v3.1	19,775	146	4250
Score v3.2	19,682	175	4314
Score v4	19,707	208	4256
Score v5	23,452	146	573
Score v6	23,456	145	570
Score v7	23,456	145	570

As we can observe in **Table 22**, with Score v1 and v2 the number of genomic variants correctly classified and misclassified was identical. When Score v3.1 was performed, the number of genomic variants correctly classified increased, because we considered the position of the negation word 'up to two words before the dictionary keyword'. The significant differences between the number of genomic variants correctly classified occurred from Score v4 to v5, *i.e.* from 19,707 to 23,452 correctly classified variants. This difference showed us that the search for 'MIM numbers' in clinical unstructured texts, was an important step to increase the number of genomic variants correctly classified. Considering the results for Score v6 the number of genomic variants correctly classified increased only by four in comparison with Score v5. Finally, the results from Score v6 were

the same for Score v7 (**Table 22**). Nevertheless, we believe that Score v7 is more suited for downstream approaches as it accounts for all occurrences of the 'MIM number'. This fact did not alter the results observed for our dataset, however could be of relevance for novel datasets used later on.

At this point, we decided to stop the manually fine-tuning underlying the previously described scores and use *Machine Learning* tools to refine our scoring approach. We expected that *Machine Learning* tools could work in an unbiased way and, by taking into account individually our dictionary keywords, could perceive which keywords were most relevant to classify a genomic variant as 'Benign' or 'Pathogenic'. After such perception, we planned to alter the individual scores of the most relevant keywords, generating a novel overall score approach and therefore increase the number of genomic variants correctly classified and decrease the number of genomic variants with an overall score of zero.

In summary, the sentiment analysis performed for each of the seven scoring strategies allowed us to understand which score was able to increase the number of genomic variants correctly classified. Therefore, we opted to use Score v7 as the starting point for the next analysis using *Machine Learning* tools.

Step 6: *Machine Learning* Approaches

The sentiment analysis was an important step to understand if the keywords in our dictionary and the corresponding overall scores could predict the classification of a genomic variant. Therefore, a genomic variant with a positive overall score would be predicted as 'Benign' while a genomic variant with a negative overall score would be predicted as 'Pathogenic'. However, the sentiment analysis was biased, with individual scores given without looking at the relevance of the keyword itself. Consequently, we opted to continue the analysis with a *Machine Learning* approach that is less unbiased and that could pinpoint which keywords were most relevant to classify a genomic variant as 'Benign' or 'Pathogenic'.

We decided to use supervised *Machine Learning* algorithms (ML), because our dataset was labelled, *i.e.* each genomic variant already contains a classification ('Benign' or 'Pathogenic'). Therefore, ML algorithms with the correct classification for each genomic variant are able to 'learn' the patterns and relationships between the keywords that best relate to the classification. Therefore, after 'learn' how to classify a genomic variant, the ML algorithms are enabled to make predictions about future genomic variants. To perform the ML approach, we followed eight steps:

- Step 6.1 Exploring and preparing the input data;
- Step 6.2 Sampling-based approaches;
- Step 6.3. Data preparation – creating random training and test datasets;
- Step 6.4 Training a model on the dataset;
- Step 6.5 Evaluating model performance;
- Step 6.6 Analysis of model overfitting;
- Step 6.7 Comparing model performance using the three distinct matrices;
- Step 6.8 Improving model performance with Random Forest method.

Step 6.1 Exploring and preparing the input data:

The classical input data for the ML approach in a context of text analysis is the 'document-term matrix'. We created three document-term matrices:

- **Frequency Matrix with Disease Frequency**: where the values into the matrix correspond to the number of times each dictionary keyword appears in the clinical unstructured text. Furthermore, we added a 'Disease' column to the matrix with the number of times any given 'MIM number' occurs in the clinical unstructured text;
- **Frequency Matrix with Disease Score**: equal to the first matrix (Frequency Matrix with Disease Frequency), however, in the 'Disease' column we added the number of times any given 'MIM number' occurs in the clinical unstructured text multiplied by -20 (a scoring approach resembling Score v7);
- **Scoring Matrix with Disease Score**: where the values into the matrix are the number of times each dictionary keyword appears in a clinical

unstructured text multiplied by the value of Score v7 calculated individually for each dictionary keyword. The ‘Disease’ column was equal to the Frequency Matrix with Disease Score.

In the three matrices, the rows correspond to each genomic variant (instances), the columns correspond to each dictionary keyword (features) and the respective negation form and the last column ‘Type’ have the two-possible classifications, ‘Benign’ or ‘Pathogenic’, for each genomic variant. For example, taking into account the following clinical unstructured text for a putative benign variant:

“This variant is considered a polymorphism in the Caucasian population and also a polymorphism in the East African population. Furthermore, due to its frequency this variant is not a mutation in both populations although studies have referred to it in the context of gastric cancer (613659) and colorectal cancer (114500).”

For this example, **Tables 23**, **24** and **25** represent the corresponding document-term matrices.

Table 23 – A representative example of the columns (features) and the ‘Type’, associated with a hypothetical ‘Benign’ variant in the Frequency Matrix with Disease Frequency.

	Frequency of Keywords Found				**Disease Frequency**	**Type**
#Variants	**polymorphism**	**mutation**	**autosomal-recessive**	**negative-mutation**		
Variant 1	2	0	0	1	2	Benign

Table 24 - A representative example of the columns (features) and the ‘Type’, associated with a hypothetical ‘Benign’ variant in the Frequency Matrix with Disease Score.

	Frequency of Keywords Found				**Disease Score**	**Type**
#Variants	**polymorphism**	**mutation**	**autosomal-recessive**	**negative-mutation**		
Variant 1	2	0	0	1	-40	Benign

Table 25 - A representative example of the columns (features) and the 'Type', associated with a hypothetical 'Benign' variant in the Scoring Matrix with Disease Score.

	Score for Keywords Found				Disease Score	Type
#Variants	**polymorphism**	**mutation**	**autosomal-recessive**	**negative-mutation**		
Variant 1	60	0	0	30	-40	Benign

The three matrices were constructed to be used separately as input in ML models. The main reason to build the three matrices was to compare the results in the ML models and understand which matrix will present the best result. The 'Scoring Matrix with Disease Score' took into account the values given by the Score v7. This matrix was very important, because it added value to the sentiment analysis. 'Benign' and 'Pathogenic' variants were kept separate for the ML approach at this stage, therefore we created three document-term matrices for each classification. For the 'Benign' classified variants, each matrix had 174 instances (*i.e.* genomic variants), 248 features (*i.e.* keywords) and one last column with the classification (target feature, *i.e.* Benign). For the 'Pathogenic' classified variants, each matrix had 23,997 instances, 248 features and one last column with the target feature, *i.e.* Pathogenic.

Step 6.2 Sampling-based approaches

The input datasets for the ML approach were the three types of document-term matrices created for the 'Benign' and 'Pathogenic' classification in the previous step (**6.1**). Of notice, we concatenated vertically both the 'Benign' and 'Pathogenic' document-term matrices, generating three master document-term matrices. Each of these master document-term matrices entailed the 24,171 instances (174 'Benign' genomic variants and 23,997 'Pathogenic' genomic variants), 255 features (dictionary keywords plus the column 'Disease' based on the presence or absent of MIM number) and the column 'Type' with the corresponding classification for each genomic variant.

These input datasets were highly imbalanced, *i.e.* each classification did not have an equal proportion of instances. To mitigate this imbalance, we used two sampling techniques: 1) the undersampling and; 2) the oversampling. With the undersampling technique, we reduced the number of instances in the majority classification, *i.e.* the classification with the highest number of instances ('Pathogenic'), until both classifications had the same number of instances. The main disadvantage for the undersampling was that this technique discards potentially useful data. With the oversampling technique we have duplicated/generated randomly the instances from the minority classification ('Benign'), until the dataset had the same number of instances for each classification. With the oversampling we avoid losing information, however we have the risk of overloading our model because we are more likely to get the same samples in training and test data, *i.e.* test dataset is no longer independent of training data. A second disadvantage of oversampling was that it increases the number of training examples, thus increasing the learning time.

To alleviate class imbalance, we used the two sampling techniques described and performed four major sampling-based approaches:

- **No action: working with the Imbalanced Input Dataset**

  For this first approach, we used the master document-term matrices as they were generated, *i.e.* without any correction of the known imbalance. It was expected that any ML model built with these highly imbalanced matrices would be biased towards the 'Pathogenic' variants, as they encompassed most instances in the document-term matrices.

- **Oversampling: Synthetic Minority Over-sampling TEchnique (SMOTE)**

  SMOTE is an over-sampling method widely used to solve ML algorithms problems involving imbalanced datasets, when the classification is binary, and one class dominates the other class in the dataset. This technique solves the class imbalance, by creating synthetic new minority instances between the existing (real) minority instances. The SMOTE is a good method to avoid the overfitting and achieves a good performance in the imbalanced data

classification problems [103]. However, the majority and minority classes can be divided into clusters and frequently, this separation is not clean and noisy samples can be generated [104]. The main reason why we did not use the SMOTE was that with this technique it is not possible to understand which instances failed the classification. The SMOTE does not provide this type of verification, because it generated new synthetic genomic variants which are not real data, *i.e.* real instances. In particular, for this Thesis, it was important to know which genomic variants were not correctly classified with the ML model, so that we could perceive and solve the problem that led to the incorrect classification.

- **Undersampling techniques:**

Two major techniques for undersampling were used: 'balanced dataset' and 'semi-balanced dataset'. With the first approach, we created 'balanced datasets' with an equal proportion of instances. Therefore, the 'balanced dataset' derived from the master document-term matrices and had 255 features (keywords) and 348 instances, *i.e.* 174 instances with 'Benign' classification (the total number of genomic variants classified as 'Benign') and 174 instances selected from the 23,997 genomic variants with 'Pathogenic' classification. This 'balanced dataset' was created considering the undersampling technique where we removed instances from the majority class ('Pathogenic'). Consequently, this sampling technique may have removed significant instances/genomic variants that could have useful information. Because of this major disadvantage, we opted not to use the 'balanced dataset' to prevent the loss of so many 'Pathogenic' variants that could be important for the ML model.

With the second undersampling technique, the 'semi-balanced datasets', we decided to create three 'semi-balanced datasets', one for each master document-term matrix. Therefore, each input dataset for the ML approach was composed by 174 instances with 'Benign' classification (the total number of genomic variants classified as 'Benign') and 1000 instances randomly selected from the total 23,997 with 'Pathogenic' classification. Therefore, the main difference between the balanced and 'semi-balanced datasets', was that

we decided to increase the number of instances selected from the majority class (‘Pathogenic’) from 174 to 1000, to diminish the amount of useful information that was removed.

After testing all these sampling techniques, we opted to use the sampling-based approach ‘semi-balanced datasets’, to create the three input matrices for the ML approach. Therefore, the Frequency Matrix with Disease Frequency, the Frequency Matrix with Disease Score and the Scoring Matrix with Disease Score were constituted by 174 instances with a ‘Benign’ classification, 1000 instances with a ‘Pathogenic’ classification (1174 lines), 255 features/keywords (255 columns) and one column with the corresponding classification (column ‘Type’).

Step 6.3. Data preparation – creating random training and test datasets:

To perform the ML approach, the three master document-term matrices were split, *i.e.* each matrix was divided into two portions: the training dataset to build the *Machine Learning* (ML) model and the testing dataset to evaluate the performance of the ML model. In order to avoid that the training and testing dataset had only genomic variants of one classification, before splitting the matrices, we mixed the vertical order of the ‘Benign’ and ‘Pathogenic’ genomic variants. The splitting was then performed, and we created the training dataset using 80% of each matrix (*i.e.* 939 genomic variants) and the testing dataset using the remaining 20% (*i.e.* 235 genomic variants). In the end, we obtained a training and a testing dataset for each of the document-term matrices.

Step 6.4 Training a model on the dataset:

In this step we decided to implement a ML method with supervised algorithm to solve our classification problem. The ML supervised algorithm selected were chosen mainly because each genomic variant in the matrices has a corresponding classification (‘Benign’ or ‘Pathogenic’), *i.e.* all the instances in the matrices were labelled. The label was used by the ML algorithm to ‘learn’ which keywords (features) were relevant to classify a genomic variant as ‘Benign’ or

‘Pathogenic’ and performed a prediction. We opted to use the Decision Tree method due to the way the knowledge acquired is displayed: with the Decision Tree method, the predictions are presented in a tree structure form, such as a flowchart, enabling an easy perception of the predictions performed without the need of any statistical measure. In the tree structure form, a genomic variant begins to be classified from the root node, *i.e.* the most important keyword, passing through several decision nodes (*i.e.* less important keywords than the root node) that are divided into branches that indicate the following decisions that can be made. In the end the final classification, *i.e.* ‘Benign’ or ‘Pathogenic’ appears in the terminal nodes. Also, important to analyse in a given Decision Tree is: 1) the depth, which is the length of the longest path from a root to a terminal node and; 2) the size, which is the number of nodes in the tree. This visual representation enables understanding exactly the set of keywords/features that were important enough to define the final classification. In particular, we have used the ‘C5.0 algorithm’ to construct our Decision Tree models that aimed to find the combination of features that best predicts the classification into ‘Benign’ or ‘Pathogenic’. We selected the ‘C5.0 algorithm’ because it is one of the most well-known algorithms as it: 1) is more efficient than other more complex models; 2) is fast to train and; 3) can be used on data with relatively few training instances resulting in a model that can be easily interpreted without mathematical background. To train our Decision Tree model we used the *C5.0* and *rpart* R packages to ascertain if with the same input the R packages generated the same results.

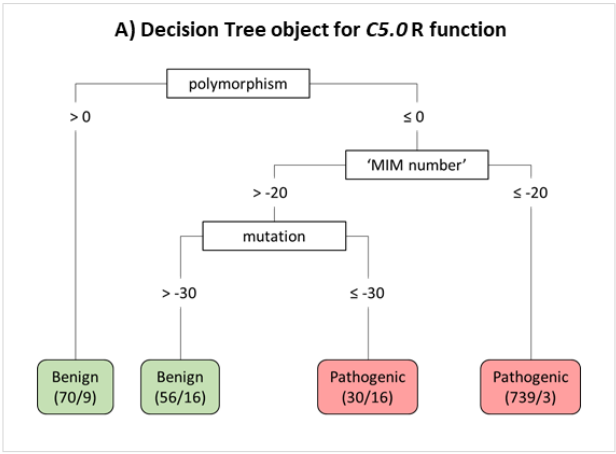
Of notice, the ML approach was made for all three matrices (Frequency Matrix with Disease Frequency, Frequency Matrix with Disease Score and Scoring Matrix with Disease Score), however for this part of the work, we have focused our observations on the results obtained for the Scoring Matrix with Disease Score. Results for the remaining matrices will be addressed in subsequent sections.

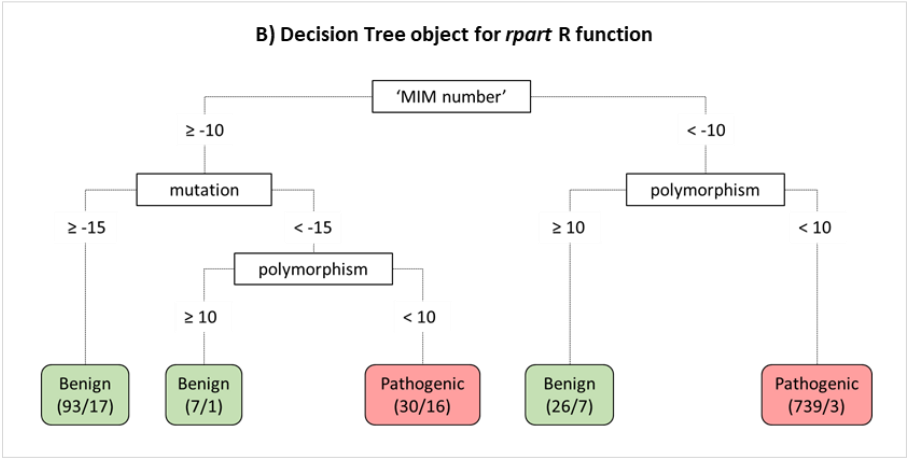
We have started by using the *C5.0* and *rpart* functions within the *C5.0* and *rpart* R packages to construct our Decision Tree model. Both functions require the features (all dictionary keywords) and the classification (column ‘Type’, *i.e.* the classification of each instance/ genomic variant) from the training dataset portion

of our matrices. This step was essential for the classifier model to ‘learn’ which features were important to distinguish between the classification, *i.e.* the 'Benign' from the 'Pathogenic' genomic variants. After the creation of the classifier model with the training dataset, a Decision Tree object was created, and a tree structure was visualized (**Figure 12A** and **12B**). In order to appease the problems that arise from the use of imbalanced datasets, as ours were, we decided to create ten distinct Decision Tree models for each training and test datasets for each matrix.

Focusing on the Decision Tree models obtained with the Scoring Matrix with Disease Score, where the individual score of each keyword was taken into account, we observed that with the *C5.0* function (**Figure 12A**), the Decision Tree model revealed only three decision nodes. These three nodes were enough to classify all the genomic variants in the corresponding training set. In particular, we observed:

1. If a clinical unstructured text had the keyword ‘polymorphism’ (> 0) the genomic variant was immediately classified as ‘Benign’;
2. Otherwise, if a clinical unstructured text did not have the keyword ‘polymorphism’ (≤ 0), the algorithm checked the value associated with the ‘MIM number’:
    - if the clinical unstructured text had one or more ‘MIM number’ (≤ -20), the genomic variant was classified as ‘Pathogenic’;
    - if a clinical unstructured text did not have a ‘MIM number’ (> -20) the algorithm next checked whether the clinical unstructured text had the keyword ‘mutation’:
        - if the keyword ‘mutation’ appeared at least once (≤ -30) the genomic variant was classified as ‘Pathogenic’;
        - otherwise, if the keyword ‘mutation’ did not appear (>-30) the genomic variant was classified as ‘Benign’.





C) Confusion Matrix for C5.0 R function

	Predicted	
Observed	Benign	Pathogenic
Benign	126 (TP)	19 (FN)
Pathogenic	25 (FP)	769 (TN)

D) Confusion Matrix for rpart R function

	Predicted	
Observed	Benign	Pathogenic
Benign	126 (TP)	19 (FN)
Pathogenic	25 (FP)	769 (TN)

Figure 13 - Comparison between results for the training dataset for *C5.0* and *rpart* R functions. Decision Tree object for *C5.0* function (**A**) and Decision Tree object for *rpart* function (**B**). For the Confusion Matrix for the training dataset performed by the *C5.0* function (**C**) and the Confusion Matrix for the training dataset performed by *rpart* function (**D**), the abbreviation TP correspond to True-Positive, TN to the True-Negative, FP to the False-Positive and FN to the False-Negative.

Of notice, not all genomic variants were correctly classified. The numbers in parentheses in the terminal node indicate the number of correctly classified and incorrectly classified variants. For example, the value '70/9' associated with the classification 'Benign' (**Figure 13A**) indicated that: 70 instances/genomic variants were correctly classified as Benign and 9 were incorrectly classified, *i.e.* these 9

instances were true ‘Pathogenic’ variants however, the Decision Tree model classified them as ‘Benign’. Considering the results in the confusion matrix for the *C5.0* function (**Figure 13C**), *i.e.* a cross-tabulation that indicates the (in)correctly classified records from the model with the training data, we observed that:

- for the genomic variants classified as ‘Benign’:

    1) 126 were true-positives, *i.e.* genomic variants correctly classified as ‘Benign’;

    2) 25 were false-positives, *i.e.* genomic variants classified as ‘Benign’, however the true classification was ‘Pathogenic’.

- for the genomic variants classified as ‘Pathogenic’:

    1) 769 true-negatives, *i.e.* genomic variants correctly classified as ‘Pathogenic’;

    2) 19 false-negatives, *i.e.* genomic variants classified as ‘Pathogenic’, however the true classification was ‘Benign’.

With the *rpart* function (**Figure 13B**), the Decision Tree model revealed 4 decision nodes, *i.e.* one more than with the *C5.0* function. Although the Decision Tree built with the rpart function involves the same dictionary keywords than the *C5.0* Decision Tree model, the order in which dictionary keywords appeared in the *rpart* Decision Tree was different. In particular, the root for the *rpart* Decision Tree was the ‘MIM number’, unlike the *C5.0* Decision Tree (‘polymorphism’). The dictionary keywords in the nodes of the Decision Tree built with the *rpart* function were the same as those in the nodes with the *C5.0* Decision Tree, however in a different order. Therefore, the decision made by the *rpart* function can be summarized in the following points:

1. If a clinical unstructured text had a ‘MIM number’ and:
    - If the dictionary keyword ‘polymorphism’ was found, then the genomic variant was classified as ‘Benign’;

- Otherwise, if the dictionary keyword ‘polymorphism’ was not found then the genomic variant was classified as ‘Pathogenic;

2. Otherwise, if the clinical unstructured text did not have a ‘MIM number’ and:
    - If the dictionary keyword ‘mutation’ was not found, then the genomic variant was classified as ‘Benign’:
    - Otherwise, if the dictionary keyword ‘mutation’ was found and:
        - The dictionary keyword ‘polymorphism’ was present, then the genomic variant was classified as ‘Benign’;
        - The dictionary keyword ‘polymorphism’ was not present, then the genomic variant was classified as ‘Pathogenic’.

Again, not all genomic variants were correctly classified. The numbers in parentheses, for example ‘93/17’ indicated that 93 instances/genomic variants were correctly classified as ‘Benign’ while 17 were incorrectly classified as ‘Benign’, *i.e.* these genomic variants were in fact ‘Pathogenic’, and the model failed. After viewing and interpreting the Decision Tree model output, we calculated the confusion matrix. As visible in **Figure 13C, D** both *C5.0* and *rpart* R functions revealed the same values in the corresponding confusion matrices for the training dataset. Therefore, both Decision Tree models correctly classified 895 of the 939 training instances/genomic variants for an error rate of 4.7%. This error rate may be overly optimistic, since Decision Tree models are known for tending to overfit, *i.e.* the model is so adjusted to the training dataset that it is not able to generalize and reliably predict new data [105]. Therefore, our next step was to evaluate our Decision Trees models using the test dataset, *i.e.* the remaining 20% of the original master matrices.

Step 6.5 Evaluating model performance

1) Accuracy

Evaluating the performance of the Decision Tree models involves applying the predict function available in the R packages. This function use the Decision Tree model and the test dataset. The confusion matrix shown in **Table 26**, considered the models built in the previous step using the test dataset previous established.

The obtained values in the confusion matrix (**Table 26**) were used to evaluate the classifier model (Decision Tree model) performance.

Table 26 - Confusion Matrix for Scoring Matrix with Disease Score for test dataset.

Test dataset	**Predicted**	
Observed	Benign	Pathogenic
Benign	24 (TP)	5 (FN)
Pathogenic	9 (FP)	197 (TN)
Accuracy	94.0%	

To evaluate the classifier model performance of a ML model, the measure of accuracy is widely used. This measure divides the proportion of correct prediction by the total number of predictions. This measure indicates the percentage of instances/genomic variants correctly or incorrectly classified by the model. The confusion matrix for the test dataset had an accuracy of 94% (**Table 26**). Although this would appear to indicate a good accurate classifier, we need to consider that the original dataset was imbalanced, *i.e.* the percentage of genomic variants classified as 'Pathogenic' in original dataset was 85% with only 15% for the 'Benign' genomic variants. Therefore, the high accuracy was related to the high number of 'Pathogenic' genomic variants in the original dataset and not because the number of genomic variants correctly classified was high for both classifications.

The imbalance between the number of instances in each classification also originated a problem in the model. The high number of genomic variants classified as 'Pathogenic' made the model 'learn' better to classify a 'Pathogenic' genomic variant than a 'Benign'. The confusion matrix proved this, since the number of misclassified true 'Benign' variants (*n*= 9, FN, **Table 26**) was higher than the number of misclassified true Pathogenic variants (*n*=5, FP, **Table 26**).

2) Precision, Recall and F1-Score

Beyond accuracy, the best measure to evaluate the performance of classifier model is whether the classifier is successful at its intended purpose. For this reason, it is important to measure the performance of a model taking into account the measure utility rather than raw accuracy. We measured the performance of the model using again the values in confusion matrix (**Table 27**). Therefore, we calculated the performance measures, precision, recall and F1-Score for the 'Benign' variants and 'negative' precision, 'negative' recall and 'negative' F1-Score for the 'Pathogenic' variants (**Table 27**).

Table 27 – Performance measures for the Scoring Matrix with Disease Score.

Performance measures related for 'Benign' variants						
Test dataset	**Precision**		**Recall**		**F1-Score**	
	C5.0	***rpart***	***C5.0***	***rpart***	***C5.0***	***rpart***
Scoring Matrix with Disease Score	0.83	0.83	0.67	0.67	0.37	0.37
Performance measures related for 'Pathogenic' variants						
Test dataset	**'Negative' Precision**		**'Negative' Recall**		**'Negative' F1-score**	
	C5.0	***rpart***	***C5.0***	***rpart***	***C5.0***	***rpart***
Scoring Matrix with Disease Score	0.94	0.94	0.98	0.98	0.48	0.48

The precision for the 'Benign' variants was calculated considered the number of 'Benign' variants correctly predicted (true-positive) divided by the total number of true 'Benign' variants in the dataset (true-positive plus false-positive). The recall also for the 'Benign' variants was calculated considered the number of 'Benign' variants correctly predicted (true-positive) divided by the total number of predicted 'Benign' variants (true-positive plus false-negatives). For the *C5.0* or *rpart* functions, the Decision Trees had a precision of 0.83 and a recall of 0.67 (**Table 27**). Both these measures are focused exclusively on true positive instances, which, in our study, were only the 'Benign' variants correctly classified.

If we calculate the same measures focusing on the true negative, *i.e.* the 'Pathogenic' variants correctly classified, we performed the 'negative' that was calculated considering the number of 'Pathogenic' variants correctly predicted (true-negative) divided by the total number of true 'Pathogenic' variants in the dataset (true-negative plus false-negative). The 'negative' recall also for the 'Pathogenic' variants was calculated considered the number of 'Pathogenic' variants correctly predicted (true-negative) divided by the total number of predicted 'Pathogenic' variants (true-negatives plus false-positives). Therefore, for the performance measures for the 'Pathogenic' variants we had from the 'negative' precision would be of 0.94 and the 'negative' recall of 0.98. By comparing these values with the previously mentioned values, this again showed that the model 'learned' better to classify a 'Pathogenic' genomic variant than a 'Benign', likely due to the imbalance of the dataset.

Another measure to evaluate the performance of the models is the F1-score that combines the precision and recall using the harmonic mean, we calculated also the 'negative' F1-score that combine the 'negative' precision and 'negative' recall. Therefore, the F1-score value obtained was 0.37 and for the 'negative' F1-score, the value was 0.45. Again, this showed that the model 'learned' better how to classify a 'Pathogenic' genomic variant than a 'Benign'.

Step 6.6 Analysis of model overfitting

Another way to evaluate the performance of the model, beyond accuracy, precision, recall and the F1-score, is the comparison between confusion matrices for the training and test dataset. This was important to perceive if there was an overfitting of the model. If the results in the confusion matrix for the training dataset were better than for the testing dataset, we can conclude that the model was so adjusted to the instances/genomic variants in the training dataset that it may not be able to generalize to new cases, *i.e.* the test dataset. Therefore, it was crucial to compare the confusion matrices for training and test datasets (**Figure 14**).

A) Confusion Matrix of Training dataset

	Training dataset	
Observed	**Predicted**	
	Benign	Pathogenic
Benign	126 (13%)	19 (2%)
Pathogenic	25 (3%)	769 (82%)
#Total variants	*n* = 939	
Accuracy	95.3%	

B) Confusion Matrix of Test dataset

	Test dataset	
Observed	**Predicted**	
	Benign	Pathogenic
Benign	24 (10%)	5 (2%)
Pathogenic	12 (5%)	194 (83%)
#Total variants	*n* = 235	
Accuracy	92.8%	

Figure 14 - Confusion Matrix of the training (**A**) and test dataset (**B**) to evaluate the presence of overfit in the model constructed with the Scoring Matrix with Disease Score.

To evaluate whether there was model overfitting, we calculated the percentage of true-positives (TP), true-negatives (TN), false-positives (FP) and false-negatives (FN) in the confusion matrices and the accuracy for each dataset (**Figure 14**). The decrease in the percentage of true-positives from 13% in training dataset to 10% in test dataset and the differences between the accuracy could be an indicator of overfitting particularly for the 'Benign' variants. These results were expected, because our master document-term matrices were imbalanced, *i.e.* only 174 'Benign' variants for 1000 'Pathogenic' variants. Nevertheless, the percentage of false-negatives did not change in both datasets (2%), showing that the model is more accurate in the classification of 'Pathogenic' variants. Altogether, these results suggest some overfitting of the model, particularly affecting 'Benign' variants.

Step 6.7 Comparing model performance using the three distinct matrices

To properly compare the three previously described matrices (Frequency Matrix with Disease Frequency, Frequency Matrix with Disease Score and Scoring Matrix with Disease Score), we next created 10 distinct Decision Tree models for each one. The main difference between the Decision Tree models were the 1000 'Pathogenic' variants used for each Decision Tree model. We randomly selected the set of 1000 'Pathogenic' variants, in the hope of decreasing the likelihood of

selecting the same sets of 'Pathogenic' variants for each model. With this, we expected to generate a more truthful model bypassing the problem of selecting only one set of 1000 'Pathogenic' variants. To better compare the results obtained, we decided to calculate the mean value for TP, TN, FP and FN obtained for each of the ten ML models developed with both the *C5.0* and *rpart* functions for each matrix. **Table 28** shows the calculated mean values for the confusion matrices obtained for the test datasets (*n* = 235 variants) created with the two R functions.

Table 28 – Comparison between the performance of the three matrices with the *C5.0* function and *rpart* function, considering the confusion matrices and accuracy for each matrix. The colours in the confusion matrices are representative of true-positive stand as green; true-negative stand as red; false-negative stand as orange; false-positive stand as blue.

Test dataset	*C5.0* function		*rpart* function		Accuracy	
					C5.0 function	*rpart* function
Frequency Matrix with Disease Frequency	26 6	8 195	24 8	6 197	**94.0%**	**94.0%**
Frequency Matrix with Disease Score	25 12	4 194	24 8	5 198	**93.2%**	**94.5%**
Scoring Matrix with Disease Score	25 12	4 194	24 9	5 197	**93.2%**	**94.0%**

Considering the results in **Table 28**, we observed that the results for the two R functions were very similar, as observed previously. Nevertheless, we observed that the *rpart* function had a higher mean value of true-negatives (19 for the *C5.0* function *vs.* 197 for the *rpart* function), which is also reflected in the accuracy results obtained. Concerning the different matrices used, the best results were obtained for the Frequency Matrix with Disease Score with the *rpart* function. Nevertheless, across matrices very similar results were observed.

Next, we calculated for all matrices the same performance measures previously described: precision, 'negative' precision, recall, 'negative' recall, F1-score and 'negative' F1-score (**Table 29**).

Table 29 – Comparison between the performance measures for the three matrices.

Performance measures related for 'Benign' variants						
Test dataset	**Precision**		**Recall**		**F1-Score**	
	C5.0	***rpart***	***C5.0***	***rpart***	***C5.0***	***rpart***
Scoring Matrix with Disease Score	0.68	0.72	0.86	0.83	0.38	0.39
Frequency Matrix with Disease Frequency	**0.81**	0.75	0.77	0.80	**0.39**	0.39
Frequency Matrix with Disease Score	0.68	0.75	0.86	0.83	0.38	0.38
Performance measures for 'Pathogenic' variants						
Test dataset	**'Negative' Precision**		**'Negative' Recall**		**'Negative' F1-score**	
	C5.0	***rpart***	***C5.0***	***rpart***	***C5.0***	***rpart***
Scoring Matrix with Disease Score	0.98	0.98	0.94	0.96	0.44	0.48
Frequency Matrix with Disease Frequency	0.96	0.97	**0.97**	0.96	**0.56**	0.48
Frequency Matrix with Disease Score	0.98	0.98	0.94	0.96	0.48	0.48

We observed that performance measures were very similar regardless of the matrix used. However, a more detailed analysis revealed that the Frequency Matrix with Disease Score had the best global performance (four best measures in a total of six measures, **Table 29**). However, this matrix cannot be further enhanced, as it recalls only keyword frequencies, unlike the Scoring Matrix with Disease Score which can be improved by altering the individual value for each keyword. To understand which the most relevant keywords were, we next analysed the tree structure built by one of the Decision Tree models for the Frequency Matrix with Disease Score (**Figure 15**).

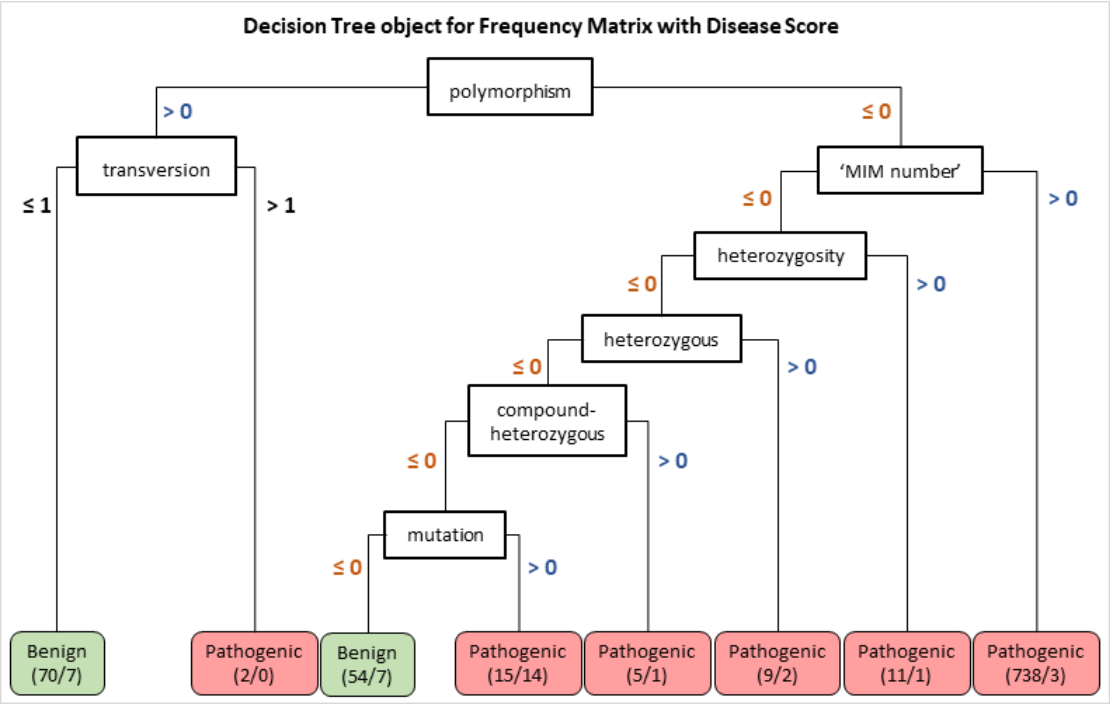


Figure 15 - Tree structure built by the Decision Tree model for the Frequency Matrix with Disease Score. The green colour is representative of 'Benign' classification and red colour for 'Pathogenic'' classification.

The tree structure built by the Decision Tree model for the Frequency Matrix with Disease Score (**Figure 15**) can be summarized in the following points:

1. If a clinical unstructured text had the dictionary keyword 'polymorphism' and:
    - If the dictionary keyword 'transversion' was not found or was found only once, the genomic variant was classified as 'Benign';
    - Otherwise, if the dictionary keyword 'transversion' was found more than once, then the genomic variant was classified as 'Pathogenic;
2. If the clinical unstructured text did not have the dictionary keyword 'polymorphism' and:
    - If a 'MIM number' was found, the genomic variant was 'Pathogenic';
    - Otherwise, if the 'MIM number' was not found and:
        - If the dictionary keyword 'heterozygosity' was found, the genomic variant was 'Pathogenic';
        - However, if the dictionary keyword 'heterozygosity' was not found and:

- If the dictionary keyword ‘heterozygous’ was found, the genomic variant was ‘Pathogenic’;
- Otherwise, if the dictionary keyword ‘heterozygous’ was not found and:
    - If the dictionary keyword ‘compound-heterozygous’ was found, the genomic variant was classified as ‘Pathogenic’;
    - However, if the dictionary keyword ‘compound-heterozygous’ was not found and:
        - If the dictionary keyword ‘mutation’ was found, the genomic variant was classified as ‘Pathogenic’;
        - Otherwise, if the dictionary keyword ‘mutation’ was not found the genomic variant was classified as ‘Pathogenic’.

From this analysis, it was possible to understand the most relevant dictionary keywords (features) in the Frequency Matrix with Disease Score which allowed the Decision Tree model to classify the genomic variants. Therefore, ‘polymorphism’ was considered relevant to classified ‘Benign’ variants and; ‘heterozygosity’, ‘heterozygous’, ‘compound-heterozygous’ and ‘mutation’ to classified ‘Pathogenic’ variants. The keyword ‘transversion’ in **Figure 15** was related with both classifications, however, we knew from the knowledge of literature that this keyword was more related with ‘Pathogenic’ variants. Taken into account the most relevant dictionary keywords observed in the tree structured, we realized that some of these dictionary keywords, such as ‘transversion’, ‘heterozygosity’ and ‘heterozygous’ had an individual score of zero, *i.e.* the individual score established in step 3 for the dictionary of biological relevant keywords was zero. Therefore, we created a next matrix, named ‘New Scoring Matrix with Disease Score’, where we altered the individual score for that keywords. For the ‘transversion’, ‘heterozygosity’ and ‘heterozygous’ that previous had an individual score of zero, in the ‘New Scoring Matrix with Disease Score’ the individual score was altered to -10, given a negative connotation for

these keywords that were decisive to classified 'Pathogenic' variants in the Decision Tree model. For the keywords 'compound-heterozygous', 'segregated', 'frameshift' and 'truncated' we increase the negativity altering the individual score from -10 to -15. This alteration was made because these keywords were directly related to 'Pathogenic' classification.

The creation of the 'New Scoring Matrix with Disease Score' allowed us to understand if an alteration in the individual score of specific keywords could increase the number of genomic variants correctly classified. Therefore, we built the ten distinct Decision Tree models with both the *C5.0* and *rpart* function for the 'New Scoring Matrix with Disease Score'. To better compare the results obtained, we decided to calculate the mean value for TP, TN, FP and FN obtained for each of the ten Decision Tree models developed with both the *C5.0* and *rpart* functions for each matrix. **Table 30** shows the calculated mean values for the confusion matrices obtained for the test datasets (*n*= 235 variants) created with the two R functions.

Table 30 - Comparison between the confusion matrices for the Scoring Matrix with Disease Score and New Scoring Matrix with Disease Score and the accuracy calculate considered the R functions *C5.0* and *rpart*.

Test dataset	***C5.0* function**		***rpart* function**		**Accuracy**	
					***C5.0* function**	***rpart* function**
Scoring Matrix with Disease Score	**24** 5	9 **197**	**24** 5	9 **197**	94.0%	94.0%
New Scoring Matrix with Disease Score	**24** 5	6 **200**	**24** 5	7 **199**	94.9%	94.8%

Considering the confusion matrices (**Table 30**) we next calculated the previously described performance measures for both the Scoring Matrix with Disease Score and New Scoring Matrix with Disease Score (**Table 30**).

Table 31 – Performance measures for the Scoring Matrix with Disease Score and New Scoring Matrix with Disease Score, calculated for the bot R function *C5.0* and *rpart*.

Performance measures related for 'Benign' variants						
Test dataset	**Precision**		**Recall**		**F1-Score**	
	C5.0	***rpart***	***C5.0***	***rpart***	***C5.0***	***rpart***
Scoring Matrix with Disease Score	0.86	0.83	0.73	0.73	0.40	0.39
New Scoring Matrix with Disease Score	0.83	0.83	**0.80**	**0.77**	**0.41**	**0.40**
Performance measures for 'Pathogenic' variants						
Test dataset	**'Negative' Precision**		**'Negative' Recall**		**'Negative' F1-score**	
	C5.0	***rpart***	***C5.0***	***rpart***	***C5.0***	***rpart***
Scoring Matrix with Disease Score	0.96	0.96	0.98	0.98	0.49	0.49
New Scoring Matrix with Disease Score	**0.97**	**0.97**	0.98	0.98	0.49	0.49

We observed that the New Scoring Matrix with Disease Score had, higher values in the performance measures (highlighted in **Table 31**). Furthermore, by comparing the confusion matrices (**Table 30**), we observed that the New Scoring Matrix with Disease Score had both a higher value of TP and TN.

Concerning the existence of overfit, as we compare the accuracy values for the New Scoring Matrix with Disease Score (95.1% for training and 95.3% for test datasets, **Figure 16A** and **16B**, respectively), we can conclude that no overfit was present. To support this, the percentage of false-negatives decreased from 3.4% (training dataset, **Figure 16B**) to 2.6% (test dataset, **Figure 16A**). Importantly, this lack of overfit observed for the Decision Tree model built using New Scoring matrix with Disease Score constituted an improvement in comparison with the

model built with the previously discussed Scoring Matrix with Disease Score (**Figure 16**).

A) Confusion Matrix of Training dataset

	Training dataset	
Observed	**Predicted**	
	Benign	Pathogenic
Benign	132 (14.1%)	32 (3.4%)
Pathogenic	14 (1.5%)	761 (81.0%)
#Total variants	*n* = 939	
Accuracy	95.1%	

B) Confusion Matrix of Test dataset

	Test dataset	
Observed	**Predicted**	
	Benign	Pathogenic
Benign	24 (10.0%)	6 (2.6%)
Pathogenic	5 (2.1%)	200 (85.3%)
#Total variants	*n* = 235	
Accuracy	95.3%	

Figure 16 - Confusion Matrix of the training (**A**) and test dataset (**B**) to evaluate the presence of overfit in the model constructed with the New Scoring Matrix with Disease Score.

Step 6.8 Improving model performance with Random Forest method

In the previous sections, we only used the Decision Tree as a *Machine Learning* method, as it was a more informative. However, we observed that the Decision Tree model was prone to overfitting, especially when a tree is particularly deep, even though the last Decision Tree model built with the New Scoring matrix with Disease Score revealed no overfit. To try to increase even more its performance, we opted to use the ensemble classifier Random Forest. A Random Forest is a collection of Decision Trees that are aggregated into a final model. Identical to the Decision Tree model, the Random Forest model 'learns' from the training dataset and makes predictions with the test dataset. The Random Forest method is considered a more robust model and is a modelling technique stronger than a single Decision Tree model.

In the Random Forest method, we used two approaches to minimize even more the overfitting of the model and increase its performance. First, the dataset used to train each Decision Tree model into the Random Forest method was unique, *i.e.* each Decision Tree was trained using a distinct set of instances. Second, not all features in the matrices were used for all Decision Trees, *i.e.* the Random Forest method only selected and used a reduced number of all available features

in each Decision Tree model. For example, if given 20 dictionary keywords, the Random Forest method may be trained using only 5 of these features. However, this could be problematic as it omits 15 features, which may be highly relevant. Nevertheless, as multiple Decision Trees are built with the Random Forest method, eventually all our original features will have been included. In the end, the classification is obtained by the voting of the classification reported by each of the different Decision Trees.

As input for the Random Forest method, we used the New Scoring Matrix with Disease Score (described in the previous section). Through the several packages available to create Random Forest method in R, the randomForest is one of the most widely used. We also used the R package caret that enables the automated fine-tuning of constructed models.

To run the *randomForest* function (within the *randomForest* package) it was required: 1) the training dataset (80% of the instances in the New Scoring Matrix with Disease Score, *i.e.* 939 genomic variants) and; 2) the classification ('Benign' or 'Pathogenic') for each instance in the training dataset. The *randomForest* and the *expand.grid* functions have parameters that can be combined to improve the model performance. In particular, the parameters considered were:

- *mtry*: Number of features randomly sampled as candidates at each Decision Tree model. For classification models, the default is the square root of the number of features. In New Scoring Matrix with Disease Score, the number of features was 255, therefore, the ideal *mtry* was 16. Nevertheless, we have also tested an *mtry* value of 128;
- *ntree*: Number of trees to grow. By default, the randomForest function creates an ensemble of 500 trees that consider *mtry* parameter. In general, more complex learning problems and larger datasets work better with larger number of trees. The goal of using a large number of trees is to train enough so that each feature has a chance to appear in several models. However, larger number of trees produce more stable models, but require more memory and longer run time. We have used *ntree* values of 1000 and 10,000.

While with the single Decision Tree model (as described in the previous sections), we can visualize the decisions underlying the tree structures, with the Random Forest method the same does not occur, as it represents a collection of several Decision Tree models. Therefore, the output object from the Random Forest method cannot be visualized. Hence, the main outputs of the Random Forest method are the confusion matrices and performance metrics. After the training stage, the Random Forest method returns an output object that was used to make predictions with the test dataset (20% of all instances, *i.e.* 235 genomic variants). To perceive which values of the parameters *mtry* and *ntree* achieve the best value of correctly classified genomic variants, we performed several rounds of Random Forest methods using different values of *mtry* and *ntree*, as shown in **Table 32**.

Table 32 – Confusion matrix and accuracy from the Random Forest methods with *mtry* 16 and *ntree* of 1000; *mtry* of 128 and *ntree* of 1000; *mtry* of 16 and of *ntree* of 10,000; *mtry* of 128 and *ntree* of 10,000.

Test Dataset	*ntree*	*mtry*	*randomForest* function		Accuracy
New Scoring Matrix with Disease Score	**1000**	**16**	**16** (6.8%)	13 (5.5%)	94.5%
			0 (0%)	**206** (87.7%)	
		128	**23** (9.8%)	6 (2.6%)	91.1%
			15 (6.4%)	**191** (81.3%)	
	10,000	**16**	**13** (5.5%)	16 (6.8%)	93.2%
			0 (0%)	**206** (87.7%)	
		128	**21** (8.9%)	4 (1.7%)	90.6%
			18 (7.7%)	**192** (81.7%)	

Considering the confusion matrices (**Table 32**), we observed that the Random Forest method with *mtry* of 16 and *ntree* of 1000, was the method with the highest accuracy (94.5%) and with the lowest values of false-positives (0) and false-negatives (13). Therefore, it appeared that a *mtry* of 16 and a *ntree* of 1000 were the parameters that best fit our dataset, as they led to the best predictions. In

particular, the best combination of correctly classified variants (16 true-positives and 206 true-negatives) and low misclassified variants (0 false-positives and 13 false-negatives). However, some considerations should be made on the different importance of false negatives and false positives. In the clinical field, it is preferable to wrongly predict a 'Benign' variant as 'Pathogenic' (false-negative) than to predict a 'Pathogenic' variant as 'Benign' (false-positive). However, *Machine Learning* methods do not consider this premise. In fact, we observed that the Random Forest method with *mtry* of 128 and *ntree* of 1000, had the lowest number of false-negatives (*n*=6) in comparison with the Random Forest method with *mtry* of 16 and *ntree* of 1000 (*n*=13). To understand more clearly the results with the Random Forest methods, we decided to calculate other performance measures (**Table 33**), beyond the accuracy.

Table 33 – Performance measures for the Random Forest methods with *mtry* 16 and *ntree* of 1000; *mtry* of 128 and *ntree* of 1000; *mtry* of 16 and *ntree* of 10,000; *mtry* of 128 and *ntree* of 10,000.

Performance measures related for 'Benign' variants					
Test Dataset	***ntree***	***mtry***	**Precision** (PPV)	**Recall** (TPR)	**F1-Score**
New Scoring Matrix with Disease Score	**1000**	**16**	1	0.55	0.35
		128	0.61	0.79	0.34
	10,000	**16**	1	0.44	0.31
		128	0.55	0.84	0.33
Performance measures for 'Pathogenic' variants					
Test Dataset	***ntree***	***mtry***	**'Negative' Precision** (NPV)	**'Negative' Recall** (TNR)	**'Negative' F1-Score'**
New Scoring Matrix with Disease Score	**1000**	**16**	0.94	1	0.48
		128	0.99	0.93	0.46
	10,000	**16**	0.93	1	0.48
		128	0.97	0.91	0.47

With the performance measures in **Table 33** we could select which Random Forest method truly best fit our dataset. For the Random Forest method with *mtry* of 16 and *ntree* of 1000 the precision value for the 'Benign' variants was 1, *i.e.*

the method correctly classifies all the ‘Benign’ variants (16 true-positives and 0 false-positives). Therefore, this model was considered very precise for the ‘Benign’ classification. However, the recall value was 0.55. This value was low in comparison with the precision value, once the method misclassified a significant number of ‘Benign’ variants (13 false-negatives). Moreover, once the precision and recall should not be analysed isolated, we further calculated the F1-Score that combines the two performance measures into a single measure. The F1-Score is the harmonic average of precision and recall and expresses how precise our model was, *i.e.* how many instances the model classified correctly and how robust it is. Therefore, with a F1-Score value of 0.35 we concluded that this Random Forest method for the ‘Benign’ variants was precise, however in terms of robustness the method was weak, once the number of ‘Benign’ variants misclassified was high. For the ‘Pathogenic’ variants in this Random Forest method (*mtry* of 16 and *ntree* of 1000), the ‘negative’ precision was 0.94, *i.e.* the method correctly classified a high number of ‘Pathogenic’ variants in the dataset (206 true-negative and 13 false-negatives). The ‘negative’ recall value was 1, once the method did not misclassify any ‘Pathogenic’ variants (0 false-positives). The ‘negative’ F1-Score was 0.48 and therefore, this method was considered as both precise and robust.

The Random Forest method with *mtry* of 128 and *ntree* of 1000 had a precision value of 0.61 for the ‘Benign’ variants, once the method had a high number of misclassified as ‘Benign’ variants (15 false-positives). The recall value was 0.79 and consequently the method was considered robust, once the number of ‘Benign’ variants misclassified as ‘Pathogenic’ (6 false-negatives) was low. The combined measure F1-Score was 0.34, hence we can conclude that the model was less precise than robust. For the ‘Pathogenic’ variants the ‘negative’ precision value was 0.99, *i.e.* the method correctly classified almost all ‘Pathogenic’ variants (191 true-negatives and 6 false-negatives). The ‘negative’ recall value was 0.93, because the method had a small number of misclassified ‘Pathogenic’ variants (15 false-positives). The ‘negative’ F1-Score’ was 0.46 and therefore this method was considered both as precise and robust.

We also built two Random Forest methods with a *ntree* parameter of 10,000 and tested two values for *mtry*: 16 and 128. For the Random Forest method with *mtry*

of 16 and a *ntree* of 10,000, the precision value for the 'Benign' variants was 1, once the method correctly classified all the 'Benign' variants (13 true-positive and 0 false-positives). The recall value was 0.44, a value considered low, since the method misclassified a significant number of 'Benign' variants (16 false-negatives). The F1-Score was 0.31 and therefore this method can be considered as precise, once it classified all correctly all the 'Benign' variants, however not truly robust. For the 'Pathogenic' variants the 'negative' precision value was 0.93, *i.e.* the method correctly classified almost all the 'Pathogenic' variants in the dataset (206 true-negatives and 16 false-negatives). The 'negative' recall value for the 'Pathogenic' variants was one given that the method did not misclassify any 'Pathogenic' variant (zero false-positives). The 'negative' F1-Score was 0.48 and therefore this method was considered both as precise and robust.

For the Random Forest method with *mtry* of 128 and *ntree* of 10,000, the precision value for 'Benign' variants was 0.55, once the method had a high number of misclassified 'Benign' variants (18 false-positives). The recall value was 0.84, once the method misclassified a low number of 'Benign' variants (4 false-negatives). The F1-Score was 0.33 and consequently the method was robust, however not very precise. For the 'Pathogenic' variants the 'negative' precision value was 0.97, as the method correctly classified almost all 'Pathogenic' variants (192 true-negatives and 4 false-negatives). The 'negative' recall value was 0.91, as the number of 'Pathogenic' variants misclassified was low (18 false-positives). Moreover, the 'negative' F1-Score was 0.31 and therefore this method was considered both precise and robust.

Considering all these four Random Forest methods (**Table 33**) and knowing that our dataset is imbalanced and that both classifications are important, we opted to choose the Random Forest method that had the highest F1-Score and 'negative' F1-score, as well as with the lowest misclassification rate. This was the case of the Random Forest method with *mtry* of 16 and *ntree* of 1000 (highlighted in **Table 33**).

After this selection, we analysed the resulting confusion matrices for both the training and the test datasets from New Scoring Matrix with Disease Score, to understand whether there was overfitting of the model. With the selected Random

Forest method (*mtry* of 16 and *ntree* of 1000), the percentage of true-positives in the training and test datasets was the same (7%), while the true-negatives went from 84.3% to 87.7% (**Figure 17A, B**). Concerning the false-positives, it improved from 8.1% in training dataset to 0% in the test dataset. This could be an indicator of no overfit in the method. However, for the false-negatives, it increased from 0.3% in the training dataset to 6% in the test dataset. Finally, we observed an increase in accuracy from 91.6% to 94.5% in the training to the test datasets. Altogether, despite the increase in false-negatives, this Random Forest model appears not to show evidences of overfitting. These results support the idea that the method was able to predict new data and therefore, our method was able to 'learn' how to distinguish between 'Pathogenic' and 'Benign' variants.

A) Confusion Matrix of Training dataset

	Training dataset	
Observed	**Predicted**	
	Benign	Pathogenic
Benign	69 (7.3%)	3 (0.3%)
Pathogenic	76 (8.1%)	791 (84.3%)
#Total variants	*n* = 939	
Accuracy	91.6%	

B) Confusion Matrix of Test dataset

	Test dataset	
Observed	**Predicted**	
	Benign	Pathogenic
Benign	16 (6.8%)	13 (5.5%)
Pathogenic	0 (0%)	206 (87.7%)
#Total variants	*n* = 235	
Accuracy	94.5%	

Figure 17 – Confusion matrices for training (**A**) and test dataset **(B**) from New Scoring Matrix Disease Score.

Although the performance of the Random Forest method was already satisfactory, we still attempted to increase its performance, by adding a resampling technique to obtain additional information on the model and thus further test it.

As previously mentioned, the *randomForest* function is also supported by the caret package in R. This package provides an excellent capability to tune *Machine Learning* parameters. In the previous section ('Improving Model Performance') we used the *ntree* and *mtry* parameters to find the Random Forest

method that best predicted our classification. However, the caret package provides other functions: in particular, we used the *trainControl* function to prepare and divide the dataset into the training and test sets using a repeated k-fold cross validation, *i.e.* 10 times 10-fold cross-validation. Therefore, with this resampling technique, the training dataset was divided into 10 subgroups (100 instances each), with nine of them used for training the model and one as a test dataset to evaluate the model performance.

We selected the same parameters for *ntree*=1000 and *mtry*=16 as described previously and choose *k*=10 for the cross-validation resampling technique. Next, we again used the *train* function to train the model, using the training dataset from New Scoring Matrix with Disease Score. The output of the *train* function was, in our case, a train Random Forest model and a table with the parameters and the performance measures. Obtained results are detailed in **Figure 18**.

To understand if the obtained Random Forest model built considering the resampling technique was able to generalize to the test dataset, we compared the accuracy and the percentage of false-positive and false-negative in both datasets. We observed that the accuracy was higher in the training dataset than in the test dataset (91.8% vs. 89.4%, **Figure 18A, B**) indicating little overfitting of the model. Concerning the percentage of false-positives and false-negatives in the training dataset (0.4% and 7.7%, **Figure 18A**), they all increased in the test dataset (1.7% and 8.9%, **Figure 18B**). This again indicated that the model had little overfit, losing some quality in the validation. Nevertheless, it can be considered a reasonably good model.

A) Confusion Matrix of Training dataset

mtry = 16 ntree = 1000 With Cross-Validation	Training dataset	
Observed	**Predicted**	
	Benign	Pathogenic
Benign	72 (7.8%)	73 (7.7%)
Pathogenic	4 (0.4%)	790 (84.1%)
#Total variants	*n* = 939	
Accuracy	91.8%	

B) Confusion Matrix of Test dataset

mtry = 16 ntree = 1000 With Cross-Validation	Test dataset	
Observed	**Predicted**	
	Benign	Pathogenic
Benign	8 (3.4%)	21 (8.9%)
Pathogenic	4 (1.7%)	202 (86.0%)
#Total variants	*n* = 235	
Accuracy	89.4%	

Figure 18 - Confusion matrices for training (A) and test dataset (B) from the Random Forest method with *mtry* of 16, *ntree* of 1000 and cross-validation from New Scoring Matrix Disease Score.

In comparison with the Random Forest method without cross-validation, we observed that the accuracy level with cross-validation was worse (94.5% vs. 89.4%, **Figure 17B**, **Figure 18B**). Therefore, applying the cross-validation resampling technique did not prove to be an advantage for our Random Forest model.

Step 6.9 Evaluating Random Forest methods performance in new data

To perform a final evaluation of the performance of the Random Forest methods built, we decided to test them with a novel dataset of clinical variants, *i.e.* 'validation dataset'. To create this validation dataset, we collected from the OMIM database 700 new clinical unstructured texts from genomic variants: 690 from 'Pathogenic' variants and 10 from 'Benign' variants. With these novel genomic variants, we created a matrix using the same scoring strategy described for the New Scoring Matrix with Disease Score, searching and scoring the same 255 dictionary keywords (features). This new matrix was used as validation dataset, to evaluate the capacity of our Random Forest method to classify variants that were never 'seen' by the method. Moreover, we built two Random Forest

methods: one without using the cross-validation technique (**Figure 19A,B**) and; another using the repeated cross-validation technique (**Figure 19C,D**). Both methods were ran with the parameters *mtry* of 16 and *ntree* of 1000.

A) Confusion Matrix of Validation-Training dataset without Cross-Validation

mtry = 16 ntree = 1000 Without Cross-Validation	Validation-Training dataset	
Observed	**Predicted**	
	Benign	Pathogenic
Benign	57 (6.1%)	89 (9.5%)
Pathogenic	2 (0.2%)	791 (84.2%)
#Total variants	*n* = 939	
Accuracy	90.3%	

B) Confusion Matrix of Validation-Test dataset without Cross-Validation

mtry = 16 ntree = 1000 Without Cross-Validation	Validation-Test dataset	
Observed	**Predicted**	
	Benign	Pathogenic
Benign	7 (1.0%)	4 (0.6%)
Pathogenic	2 (0.3%)	687 (98.1%)
#Total variants	*n* = 700	
Accuracy	99.2%	

C) Confusion Matrix of Validation-Training dataset with Cross-Validation

mtry = 16 ntree = 1000 With Cross-Validation	Validation-Training dataset	
Observed	**Predicted**	
	Benign	Pathogenic
Benign	60 (6.4%)	86 (9.2%)
Pathogenic	2 (0.2%)	791 (84.2%)
#Total variants	*n* = 939	
Accuracy	90.6%	

D) Confusion Matrix of Validation-Test dataset with Cross-Validation

mtry = 16 ntree = 1000 With Cross-Validation	Validation-Test dataset	
Observed	**Predicted**	
	Benign	Pathogenic
Benign	7 (1.0%)	4 (0.6%)
Pathogenic	3 (0.4%)	686 (98.0%)
#Total variants	*n* = 700	
Accuracy	99.0%	

Figure 19 - Confusion matrix for the validation-training dataset built with Random Forest with *mtry* of 16, *ntree* of 1000 and without cross-validation (**A**) and the confusion matrix for validation-test dataset created the 700 novel genomic variants (**B**). Confusion matrix performed with the Random Forest method with *mtry* of 16, *ntree* of 1000 and with cross-validation (**C**) and the confusion matrix to evaluate the performance of the method considering the 700 novel genomic variants of the validation dataset **(D)**.

With the Random Forest method without cross-validation technique, the accuracy value increased from 90.3% in the training dataset to 99.2% in the test dataset (**Figure 19A, B**). We observed also that the percentage of false-positives did not vary much: 0.2% in the training dataset and 0.3% in the test dataset. The percentage of false-negatives was vastly improved, from 9.5% in the training dataset to 0.6% in the test dataset (**Figure 19A, B**). These results demonstrate that overfit did not occur when the Random Forest method without cross-validation was applied to a completely new set of genomic variants.

The Random Forest method with cross-validation technique, had an accuracy value of 90.6% in the training dataset (**Figure 19C**) and 99.0% for the test dataset (**Figure 19C, D**). The percentage of false-positives was similar in both datasets (0.2% in training dataset and 0.4% in test dataset) and the percentage of false-negatives decreased from 9.2% in training dataset to 0.6% in the test dataset (**Figure 19C, D**). Therefore, considering these results we concluded that overfit did not occur when this Random Forest method.

Next, we calculated the performance measures for the test datasets of both methods, to evaluate their performance with and without the cross-validation technique (**Table 34**). The Random Forest method without cross-validation, for the 'Benign' variants had the precision value of 0.78, *i.e.* the method correctly classified a significant number of 'Benign' variants (7 true-positives and 2 false-positives). The recall value for the 'Benign' variants was 0.64, therefore the method misclassified few 'Benign' variants (4 false-negatives). The F1-Score was 0.35 and the method was considered more precise than robust. For the 'Pathogenic' variants the value of 'negative' precision and 'negative' recall was 0.99 both, once the method correctly predicted almost all 'Pathogenic' variants (687 true-negative, 4 false-negatives and 2 false-positives). The F1-Score was 0.50. Therefore, this method was considered precise and robust (**Table 34**).

Considering the performance measures for the Random Forest method with cross-validation (**Table 34**), the precision value for the 'Benign' variants was 0.64, *i.e.* the method correctly classified 7 'Benign' variants in a total of 10 'Benign' variants in the test dataset (7 true-positives). The recall value was 0.70, the method misclassified only 4 'Benign' variants (4 false-negatives). The F1-Score

was 0.33. Therefore, the model was considered more robust than precise. For the 'Pathogenic' variants, the 'negative' precision was 0.99, once the method almost correctly predicted all the 'Pathogenic' variants (686 true-negatives). The 'negative' recall was 1, *i.e.* the method had few misclassified 'Pathogenic' variants (3 false-positives, **Table 34**). The F1-Score was 0.50 and the method was considered as precise than robust.

Table 34 – Performance measures for the Random Forest methods with *mtry* of 16 and *ntree* of 1000 and with/without cross-validation technique.

Performance measures related for 'Benign' variants						
Validation-Test Dataset	***ntree***	***mtry***	**Resampling Tecnhique**	**Precision (PPV)**	**Recall (TPR)**	**F1-Score**
Novel genomic variants (*n*= 700)	**1000**	**16**	**Without Cross-Validation**	0.78	0.64	0.35
			With Cross-Validation	0.64	0.70	0.33
Performance measures for 'Pathogenic' variants						
Validation-Test Dataset	***ntree***	***mtry***	**Resampling Tecnhique**	**'Negative' Precision (NPV)**	**'Negative' Recall (TNR)**	**'Negative' F1-Score'**
Novel genomic variants (*n*= 700)	**1000**	**16**	**Without Cross-Validation**	0.99	0.99	0.50
			With Cross-Validation	0.99	1	0.50

Comparing the confusion matrices, the accuracies and the performance measures for both Random Forest methods, we concluded, for our case, despite the imbalance of the dataset, the cross-validation technique did not improve the good results already obtain without cross-validation.

In conclusion, our Random Forest model (with or without cross-validation) was able to learn from the new dataset of genomic variants and predict the classification of the test dataset with high precision and recall.

Conclusion and Future Perspectives

Next Generation Sequencing has allowed the detection of a large number of genomic variants associated with disease. Therefore, with this technology and the appropriated bioinformatics tools and clinical databases, the amount of data related to the functional impact of detected variants has increased in an exponential way in the last years. However, the clinical information related with the genomic variants are often stored in databases where such information is concealed within unstructured texts. The Online Mendelian Inheritance in Man database, OMIM, is one of the largest databases where clinical information of genomic variants is stored in unstructured texts. However, unlike other databases such as ClinVar, OMIM does not provide a clear classification, such as 'Pathogenic' or 'Benign' of the deposited genomic variants. Although correspondences between OMIM and ClinVar can be established, as many genomic variants are present in both databases, many other genomic variants described in OMIM are either unclassified in ClinVar or classified as 'variants of unknown significance' (VUS), a classification without clinical certainty. In light of this, the aim of this Thesis was to develop a tool, combining *Text Mining* and *Machine Learning* approaches, that allows the extraction of information from OMIM clinical description towards the classification of a genomic variant as 'Benign' or as 'Pathogenic', with a given certainty.

To develop this tool, we have used the unstructured clinical texts from OMIM as input and performed several pre-processing steps for removal of unnecessary information such as punctuation. Afterwards, we created a dictionary of clinically relevant keywords with different connotations and biological implications, selected based on the knowledge from the literature. Using *Text Mining* and the *Machine Learning* methods Decision Tree and Random Forest, we were able to build a prediction method with a high value of confidence that can predict the classification 'Benign' or 'Pathogenic' for novel genomic variants. Using a dataset of genomic variants distinct from that used for the development of the tool, we were able to obtain an accuracy rate of 99% in the prediction of pathogenicity.

Considering these very good results, our future perspectives are to test and apply our tool for the classification with a given certainty of genomic variants that are

currently classified as Variants of Unknown Significance (VUS) in ClinVar. Moreover, our tool could also be adapted, by adding novel keywords to our dictionary, for the analysis of clinical unstructured texts of genomic variants classified as, for example, 'Likely Pathogenic', 'Likely Benign' and 'Drug Response'. Finally, we also aim to adapt our tool to the interpretation and classification of unstructured texts from other databases where relevant information concerning genomic variants is also concealed, such as UniProt.

In conclusion, using a combination of *Text Mining* and *Machine Learning* approaches, we were able to create an accurate, robust and precise tool that is capable of interpreting clinical unstructured texts and to confidently assess the pathogenicity of genomic variants.

References

[1] «dbSNP and dbVar: NCBI Databases of Simple and Structural Variations». [Em linha]. Disponível em: https://www.ncbi.nlm.nih.gov/variation/dbSNP_dbVar_FAQ/. [Acedido: 29-Dez-2017].
[2] U. Farooq, H. Mansoor, A. Nongaillard, Y. Ouzrout, e M. A. Qadir, *Negation Handling in Sentiment Analysis at Sentence Level*, vol. 12. 2016.
[3] R. Nielsen, J. S. Paul, A. Albrechtsen, e Y. S. Song, «Genotype and SNP calling from next-generation sequencing data», *Nat. Rev. Genet.*, vol. 12, n. 6, pp. 443–451, Jun. 2011.
[4] R. Feldman e J. Sanger, *THE TEXT MINING HANDBOOK - Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, 2007.
[5] R. Kohavi e F. Provost, *Glossary of terms. Machine Learning*. 1998.
[6] G. Wunnava, «Applying Machine Learning to Text Mining». Jun-2015.
[7] F. Zhu *et al.*, «Biomedical text mining and its applications in cancer research», *J. Biomed. Inform.*, vol. 46, n. 2, pp. 200–211, Abr. 2013.
[8] C. C. Chrystoja e E. P. Diamandis, «Whole Genome Sequencing as a Diagnostic Test: Challenges and Opportunities», *Clin. Chem.*, vol. 60, n. 5, pp. 724–733, Mai. 2014.
[9] A. B. Singleton, «Exome sequencing: a transformative technology», *Lancet Neurol.*, vol. 10, n. 10, pp. 942–946, Out. 2011.
[10] K. Lohmann e C. Klein, «Next Generation Sequencing and the Future of Genetic Diagnosis», *Neurotherapeutics*, vol. 11, n. 4, pp. 699–707, Out. 2014.
[11] F. Sanger, S. Nicklen, e A. R. Coulson, «DNA sequencing with chain-terminating inhibitors», *Proc. Natl. Acad. Sci. U. S. A.*, vol. 74, n. 12, pp. 5463–5467, Dez. 1977.
[12] A. Grada e K. Weinbrecht, «Next-generation sequencing: methodology and application», *J. Invest. Dermatol.*, vol. 133, n. 8, p. e11, Ago. 2013.
[13] L. Liu *et al.*, «Comparison of Next-Generation Sequencing Systems», *J. Biomed. Biotechnol.*, vol. 2012, pp. 1–11, 2012.
[14] D. Peakall e L. Shugart, «The Human Genome Project (HGP)», *Ecotoxicol. Lond. Engl.*, vol. 11, n. 1, p. 7, Fev. 2002.
[15] L. Technologies Corporation, «Ion Torrent: Amplicon Sequencing». 2011.
[16] C. S. Pareek, R. Smoczynski, e A. Tretyn, «Sequencing technologies and genome sequencing», *J. Appl. Genet.*, vol. 52, n. 4, pp. 413–435, Nov. 2011.
[17] Y. Feng, Y. Zhang, C. Ying, D. Wang, e C. Du, «Nanopore-based fourth-generation DNA sequencing technology», *Genomics Proteomics Bioinformatics*, vol. 13, n. 1, pp. 4–16, Fev. 2015.
[18] W. Timp, U. M. Mirsaidov, D. Wang, J. Comer, A. Aksimentiev, e G. Timp, «Nanopore Sequencing: Electrical Measurements of the Code of Life», *IEEE Trans. Nanotechnol.*, vol. 9, n. 3, pp. 281–294, Mai. 2010.
[19] S. S. Amr e B. Funke, «Chapter 16 - Targeted Hybrid Capture for Inherited Disease Panels», em *Clinical Genomics*, S. Kulkarni e J. Pfeifer, Eds. Boston: Academic Press, 2015, pp. 251–269.
[20] J. K. Teer e J. C. Mullikin, «Exome sequencing: the sweet spot before whole genomes», *Hum. Mol. Genet.*, vol. 19, n. R2, pp. R145–R151, Out. 2010.
[21] Q. Wang, C. S. Shashikant, M. Jensen, N. S. Altman, e S. Girirajan, «Novel metrics to measure coverage in whole exome sequencing datasets reveal local and global non-uniformity», *Sci. Rep.*, vol. 7, n. 1, Dez. 2017.

[22] L. Bromham, *An Introduction to Molecular Evolution and Phylogenetics*, Second Edition. Oxford, New York: Oxford University Press, 2016.
[23] H. Yang, D. Li, e C. Cheng, «Relating gene expression evolution with CpG content changes», *BMC Genomics*, vol. 15, p. 693, Ago. 2014.
[24] E. T. Kool, «Hydrogen bonding, base stacking, and steric effects in dna replication», *Annu. Rev. Biophys. Biomol. Struct.*, vol. 30, pp. 1–22, 2001.
[25] R. Tewhey *et al.*, «Enrichment of sequencing targets from the human genome by solution hybridization», *Genome Biol.*, vol. 10, n. 10, p. R116, 2009.
[26] F. Mertes *et al.*, «Targeted enrichment of genomic DNA regions for next-generation sequencing», *Brief. Funct. Genomics*, vol. 10, n. 6, pp. 374–386, Nov. 2011.
[27] K. Kim *et al.*, «Effect of Next-Generation Exome Sequencing Depth for Discovery of Diagnostic Variants», *Genomics Inform.*, vol. 13, n. 2, pp. 31–39, Jun. 2015.
[28] M. J. P. Chaisson, R. K. Wilson, e E. E. Eichler, «Genetic variation and the de novo assembly of human genomes», *Nat. Rev. Genet.*, vol. 16, n. 11, pp. 627–640, Nov. 2015.
[29] C. Hughes, B. Ma, e G. A. Lajoie, «De novo sequencing methods in proteomics», *Methods Mol. Biol. Clifton NJ*, vol. 604, pp. 105–121, 2010.
[30] F. Moraes e A. Góes, «A decade of human genome project conclusion: Scientific diffusion about our genome knowledge: A Decade of Human Genome Project Conclusion», *Biochem. Mol. Biol. Educ.*, vol. 44, n. 3, pp. 215–223, Mai. 2016.
[31] R. M. Durbin *et al.*, «A map of human genome variation from population-scale sequencing», *Nature*, vol. 467, n. 7319, pp. 1061–1073, Out. 2010.
[32] A. Auton *et al.*, «A global reference for human genetic variation», *Nature*, vol. 526, n. 7571, pp. 68–74, Set. 2015.
[33] «The International HapMap Project | Nature». [Em linha]. Disponível em: https://www.nature.com/articles/nature02168. [Acedido: 24-Out-2018].
[34] World Health Organization, «WHO | Cancer», *WHO*. [Em linha]. Disponível em: http://www.who.int/mediacentre/factsheets/fs297/en/. [Acedido: 27-Jan-2018].
[35] C. S. Ku, E. Y. Loy, A. Salim, Y. Pawitan, e K. S. Chia, «The discovery of human genetic variations and their use as disease markers: past, present and future», *J. Hum. Genet.*, vol. 55, n. 7, pp. 403–415, Jul. 2010.
[36] M. T. Maurano *et al.*, «Systematic Localization of Common Disease-Associated Variation in Regulatory DNA», *Science*, vol. 337, n. 6099, pp. 1190–1195, Set. 2012.
[37] L. Chen, P. Jin, e Z. S. Qin, «DIVAN: accurate identification of non-coding disease-specific risk variants using multi-omics profiles», *Genome Biol.*, vol. 17, n. 1, Dez. 2016.
[38] Y. G. Tak e P. J. Farnham, «Making sense of GWAS: using epigenomics and genome engineering to understand the functional relevance of SNPs in non-coding regions of the human genome», *Epigenetics Chromatin*, vol. 8, n. 1, Dez. 2015.
[39] A. C. Nica e E. T. Dermitzakis, «Expression quantitative trait loci: present and future», *Philos. Trans. R. Soc. B Biol. Sci.*, vol. 368, n. 1620, Jun. 2013.
[40] T. R. Bhangale, M. J. Rieder, R. J. Livingston, e D. A. Nickerson, «Comprehensive identification and characterization of diallelic insertion–deletion polymorphisms in 330 human candidate genes», *Hum. Mol. Genet.*, vol. 14, n. 1, pp. 59–69, Jan. 2005.

[41] R. E. Mills, «An initial map of insertion and deletion (INDEL) variation in the human genome», *Genome Res.*, vol. 16, n. 9, pp. 1182–1190, Ago. 2006.
[42] C. Stamoulis, «Estimation of correlations between copy-number variants in non-coding DNA», 2011, pp. 5563–5566.
[43] P. Stankiewicz e J. R. Lupski, «Structural Variation in the Human Genome and its Role in Disease», *Annu. Rev. Med.*, vol. 61, n. 1, pp. 437–455, Fev. 2010.
[44] E. Gonzalez, «The Influence of CCL3L1 Gene-Containing Segmental Duplications on HIV-1/AIDS Susceptibility», *Science*, vol. 307, n. 5714, pp. 1434–1440, Mar. 2005.
[45] K. Fellermann *et al.*, «A Chromosome 8 Gene-Cluster Polymorphism with Low Human Beta-Defensin 2 Gene Copy Number Predisposes to Crohn Disease of the Colon», *Am. J. Hum. Genet.*, vol. 79, n. 3, pp. 439–448, Set. 2006.
[46] L. Feuk, A. R. Carson, e S. W. Scherer, «Structural variation in the human genome», *Nat. Rev. Genet.*, vol. 7, n. 2, pp. 85–97, Fev. 2006.
[47] M. Sayitoğlu, «Clinical Interpretation of Genomic Variations», *Turk. J. Hematol.*, vol. 33, n. 3, pp. 172–179, Ago. 2016.
[48] K. D. Pruitt *et al.*, «RefSeq: an update on mammalian reference sequences», *Nucleic Acids Res.*, vol. 42, n. D1, pp. D756–D763, Jan. 2014.
[49] B. M. Good, E. L. Clarke, L. de Alfaro, e A. I. Su, «The Gene Wiki in 2011: community intelligence applied to human gene annotation», *Nucleic Acids Res.*, vol. 40, n. D1, pp. D1255–D1261, Jan. 2012.
[50] D. A. Benson, K. Clark, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, e E. W. Sayers, «GenBank», *Nucleic Acids Res.*, vol. 42, n. D1, pp. D32–D37, Jan. 2014.
[51] C. Brooksbank, M. T. Bergman, R. Apweiler, E. Birney, e J. Thornton, «The European Bioinformatics Institute's data resources 2014», *Nucleic Acids Res.*, vol. 42, n. D1, pp. D18–D25, Jan. 2014.
[52] S. T. Sherry *et al.*, «dbSNP: the NCBI database of genetic variation», *Nucleic Acids Res.*, vol. 29, n. 1, pp. 308–311, Jan. 2001.
[53] G. A. McVean *et al.*, «An integrated map of genetic variation from 1,092 human genomes», *Nature*, vol. 491, n. 7422, pp. 56–65, Out. 2012.
[54] A. Kiran e P. V. Baranov, «DARNED: a DAtabase of RNa EDiting in humans», *Bioinformatics*, vol. 26, n. 14, pp. 1772–1776, Jul. 2010.
[55] The UniProt Consortium, «UniProt: a hub for protein information», *Nucleic Acids Res.*, vol. 43, n. D1, pp. D204–D212, Jan. 2015.
[56] S. A. Forbes *et al.*, «COSMIC: exploring the world's knowledge of somatic mutations in human cancer», *Nucleic Acids Res.*, vol. 43, n. D1, pp. D805–D811, Jan. 2015.
[57] A. Hamosh, «Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders», *Nucleic Acids Res.*, vol. 33, n. Database issue, pp. D514–D517, Dez. 2004.
[58] M. J. Landrum *et al.*, «ClinVar: public archive of relationships among sequence variation and human phenotype», *Nucleic Acids Res.*, vol. 42, n. D1, pp. D980–D985, Jan. 2014.
[59] «The UniProt databases», *EMBL-EBI Train online*, 28-Nov-2013. [Em linha]. Disponível em: https://www.ebi.ac.uk/training/online/course/uniprot-quick-tour/what-uniprot/uniprot-databases-0. [Acedido: 29-Dez-2017].

[60] J. S. Amberger, C. A. Bocchini, F. Schiettecatte, A. F. Scott, e A. Hamosh, «OMIM.org: Online Mendelian Inheritance in Man (OMIM®), an online catalog of human genes and genetic disorders», *Nucleic Acids Res.*, vol. 43, n. D1, pp. D789–D798, Jan. 2015.

[61] J. Pevsner, *Bioinformatics and Functional Genomics*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2009.

[62] C. D. Bajdik, B. Kuo, e S. Rusaw, «CGMIM: Automated text-mining of Online Mendelian Inheritance in Man (OMIM) to identify genetically-associated cancers and candidate genes», Mar. 2005.

[63] V. A. McKusick, «Mendelian Inheritance in Man and Its Online Version, OMIM», *Am. J. Hum. Genet.*, vol. 80, n. 4, pp. 588–604, Abr. 2007.

[64] A. Kitts, L. Phan, Ward, e Bradley Holmes, «The Database of Short Genetic Variation (dbSNP)», em *The NCBI Handbook [Internet]. 2nd edition.*, 2nd edition., 2014.

[65] A. Kitts, D. Church, T. Hefferon, e L. Phan, «Database of Genomic Structural Variation (dbVar)», em *The NCBI Handbook [Internet].*, 2ªedition., 2014.

[66] S. Richards *et al.*, «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», *Genet. Med. Off. J. Am. Coll. Med. Genet.*, vol. 17, n. 5, pp. 405–424, Mai. 2015.

[67] «The Seven Practice Areas of Text Analytics», em *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*, Elsevier, 2012, pp. 29–41.

[68] P. M. Nadkarni, «An introduction to information retrieval: applications in genomics», *Pharmacogenomics J.*, vol. 2, n. 2, pp. 96–102, 2002.

[69] L. Kumar e P. K. Bhatia, «TEXT MINING: CONCEPTS, PROCESS AND APPLICATIONS», *J. Glob. Res. Comput. Sci.*, vol. 4, n. 3, pp. 36–39, Jan. 1970.

[70] V. Gurusamy e S. Kannan, «Preprocessing Techniques for Text Mining», 2014.

[71] R. Talib, M. K. Hanif, e S. Ayesha, «Text Mining: Techniques, Applications and Issues», 2016.

[72] A. Kao e S. R. Poteet, *Natural Language Processing and Text Mining*. Springer Science & Business Media, 2007.

[73] F. Zhu *et al.*, «Biomedical text mining and its applications in cancer research», *J. Biomed. Inform.*, vol. 46, n. 2, pp. 200–211, Abr. 2013.

[74] S. Raschka e V. Mirjalili, *Python machine learning: machine learning and deep learning with Python, scikit-learn, and TensorFlow*, Second edition. Birmingham Mumbai: Packt Publishing Ltd, 2017.

[75] W. B. Cavnar e J. M. Trenkle, «N-Gram-Based Text Categorization», em *In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, 1994, pp. 161–175.

[76] R. Irfan *et al.*, «A survey on text mining in social networks», *Knowl. Eng. Rev.*, vol. 30, n. 02, pp. 157–170, Mar. 2015.

[77] B. Lantz, *Machine learning with R: learn how to use R to apply powerful machine learning methods and gain an insight into real-world applications*. Birmingham: Packt Publ, 2013.

[78] G. Hripcsak e A. S. Rothschild, «Agreement, the F-Measure, and Reliability in Information Retrieval», *J. Am. Med. Inform. Assoc. JAMIA*, vol. 12, n. 3, pp. 296–298, 2005.
[79] G. James, D. Witten, T. Hastie, e R. Tibshirani, *An introduction to statistical learning: with applications in R*, Corrected at 8th printing. New York Heidelberg Dordrecht London: Springer, 2017.
[80] J. Davis e M. Goadrich, «The relationship between Precision-Recall and ROC curves», em *Proceedings of the 23rd international conference on Machine learning - ICML '06*, Pittsburgh, Pennsylvania, 2006, pp. 233–240.
[81] U. Fayyad, G. Piatetsky-Shapiro, e P. Smyth, «From data mining to knowledge discovery in databases», 1996.
[82] S. Nam e T. Park, «Pathway-Based Evaluation in Early Onset Colorectal Cancer Suggests Focal Adhesion and Immunosuppression along with Epithelial-Mesenchymal Transition», *PLoS ONE*, vol. 7, n. 4, p. e31685, Abr. 2012.
[83] J. Li, X. Zhu, e J. Y. Chen, «Building Disease-Specific Drug-Protein Connectivity Maps from Molecular Interaction Networks and PubMed Abstracts», *PLoS Comput. Biol.*, vol. 5, n. 7, p. e1000450, Jul. 2009.
[84] J. M. Fernandez, R. Hoffmann, e A. Valencia, «iHOP web services», *Nucleic Acids Res.*, vol. 35, n. Web Server, pp. W21–W26, Mai. 2007.
[85] N. Papanikolaou *et al.*, «BioTextQuest + : a knowledge integration platform for literature mining and concept discovery», *Bioinformatics*, vol. 30, n. 22, pp. 3249–3256, Nov. 2014.
[86] H.-W. Chun *et al.*, «Automatic recognition of topic-classified relations between prostate cancer and genes using MEDLINE abstracts», *BMC Bioinformatics*, vol. 7, n. Suppl 3, p. S4, Nov. 2006.
[87] X. Deng, H. Geng, D. R. Bastola, e H. H. Ali, «Link test--A statistical method for finding prostate cancer biomarkers», *Comput. Biol. Chem.*, vol. 30, n. 6, pp. 425–433, Dez. 2006.
[88] M. Krallinger, F. Leitner, e A. Valencia, «Analysis of biological processes and diseases using text mining approaches», *Methods Mol. Biol. Clifton NJ*, vol. 593, pp. 341–382, 2010.
[89] S. Bird, E. Klein, e E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*, 1. ed. Sebastopol, Calif.: O'Reilly, 2009.
[90] «Inflection — inflection 0.3.1 documentation». [Em linha]. Disponível em: https://inflection.readthedocs.io/en/latest/. [Acedido: 24-Out-2018].
[91] K. Welbers, W. Van Atteveldt, e K. Benoit, «Text Analysis in R», *Commun. Methods Meas.*, vol. 11, n. 4, pp. 245–265, Out. 2017.
[92] K. L. Sainani, «Understanding Odds Ratios», *PM&R*, vol. 3, n. 3, pp. 263–267, Mar. 2011.
[93] I. Rajman, L. Knapp, T. Morgan, e C. Masimirembwa, «African Genetic Diversity: Implications for Cytochrome P450-mediated Drug Metabolism and Drug Development», *EBioMedicine*, vol. 17, pp. 67–74, Mar. 2017.
[94] L. Lennard, «Implementation of TPMT testing», *Br. J. Clin. Pharmacol.*, vol. 77, n. 4, pp. 704–714, Abr. 2014.
[95] J. O. Miners e D. J. Birkett, «Cytochrome P4502C9: an enzyme of major importance in human drug metabolism», *Br. J. Clin. Pharmacol.*, vol. 45, n. 6, pp. 525–538, Jun. 1998.

[96] R. H. N. van Schaik, «6. Dose Adjustments Based on Pharmacogenetics of CYP450 Enzymes», *EJIFCC*, vol. 19, n. 1, pp. 42–47, Abr. 2008.
[97] S. A. Tishkoff *et al.*, «The Genetic Structure and History of Africans and African Americans», *Science*, vol. 324, n. 5930, pp. 1035–1044, Mai. 2009.
[98] P. S e S. F. S, «Opinion Mining and Sentiment Analysis - An Assessment of Peoples' Belief: A Survey», *Int. J. Ad Hoc Sens. Ubiquitous Comput.*, vol. 4, n. 1, pp. 21–33, Fev. 2013.
[99] «Text Mining: Creating Tidy Text · UC Business Analytics R Programming Guide». [Em linha]. Disponível em: https://uc-r.github.io/tidy_text. [Acedido: 24-Out-2018].
[100] F. Å. Nielsen, «A new ANEW: Evaluation of a word list for sentiment analysis in microblogs», *ArXiv11032903 Cs*, Mar. 2011.
[101] L. Zhang, S. Wang, e B. Liu, «Deep learning for sentiment analysis: A survey», *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, n. 4, p. e1253, Jul. 2018.
[102] «NRC Emotion Lexicon». [Em linha]. Disponível em: https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm. [Acedido: 24-Out-2018].
[103] Y. Mi, «Imbalanced Classification Based on Active Learning SMOTE», *Res. J. Appl. Sci. Eng. Technol.*, vol. 5, n. 3, pp. 944–949, Jan. 2013.
[104] Haibo He e E. A. Garcia, «Learning from Imbalanced Data», *IEEE Trans. Knowl. Data Eng.*, vol. 21, n. 9, pp. 1263–1284, Set. 2009.
[105] L. Rokach e O. Maimon, *Data mining with decision trees: theory and applications*. Singapore ; Hackensack, NJ: World Scientific, 2008.