POLITECNICO
MILANO 1863

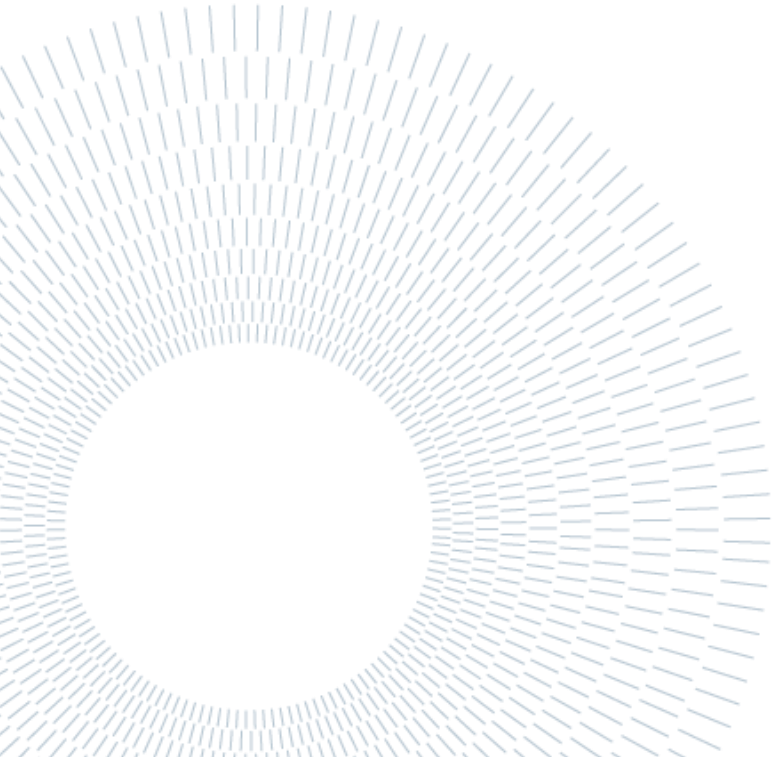
SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Identification of RAS co-occurrent mutations in colorectal cancer patients: workflow assessment and enhancement

TESI DI LAUREA MAGISTRALE IN BIOMEDICAL ENGINEERING - INGEGNERIA BIOMEDICA

Author: **Maria Laura Chieruzzi**

Student ID: 971407
Advisor: Marco Masseroli
Co-advisor: Silvia Cascianelli
Academic Year: 2021-2022



Abstract

Next Generation Sequencing (NGS) technology has made it possible, in recent decades, to obtain a lot of mutational data in a short time and at a low cost. Based on the wealth of mutational data generated NGS technology, this thesis focuses on colorectal cancer patients (CRC) with mutations in the RAS gene family (KRAS, NRAS, HRAS), since these are patients who, unfortunately, do not respond to conventional therapies.

Our main objective is the identification in RAS-mutated CRC patients of co-occurrent mutations, whose actionability may be further investigated. The study aims to enhance a previous workflow developed for identifying the most frequently co-occurring mutated genes in a RAS-mutated subpopulation of CRC patients. Given the promising results of that work, this thesis project aims to improve the encoding and selection phases, needed to transform the available mutational data into relevant features for Machine Learning techniques. Additionally, it seeks to optimize the prediction phase and its outcomes by evaluating and improving the previously proposed Data Science-based pipeline. The objective is to identify the most effective strategy in terms of performance and highlight the relevant features in the proposed methods. The resulting relevant features could thus be a starting point for personalized therapies for CRC patients who do not respond to conventional therapies.

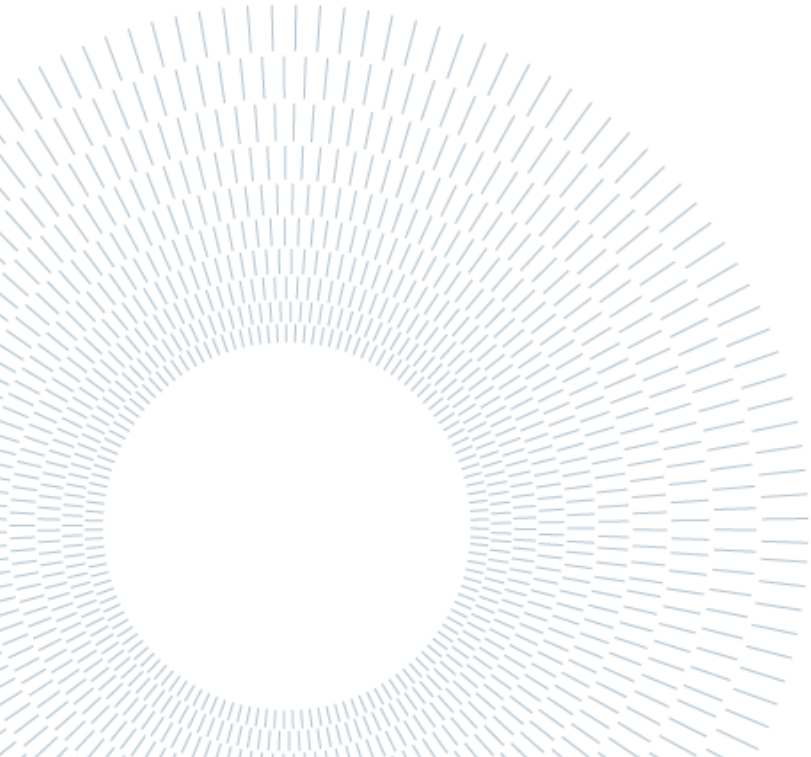
Key words: RAS gene family, Machine Learning, mutations, feature selection, bioinformatics, colorectal cancer, encoding, computational genomics.

Abstract in italiano

La tecnologia Next Generation Sequencing (NGS) ha reso possibile, negli ultimi decenni, ottenere una grande quantità di dati mutazionali in breve tempo e a basso costo. Sulla base della ricchezza di dati mutazionali generati dalla tecnologia NGS, questa tesi si concentra sui pazienti affetti da cancro del colon-retto (CRC) con mutazioni nella famiglia dei geni RAS (KRAS, NRAS, HRAS), poiché questi pazienti, purtroppo, non rispondono alle terapie convenzionali.

Il nostro obiettivo principale è l'identificazione, nei pazienti affetti da CRC con mutazioni RAS, di mutazioni co-occorrenti, le cui influenzabilità devono essere ulteriormente investigate. Lo studio mira a migliorare il flusso di lavoro precedente sviluppato per identificare i geni mutati più frequentemente co-occorrenti in una sotto-popolazione di pazienti affetti da CRC con mutazioni RAS. Dati i promettenti risultati di quel lavoro, questo progetto di tesi mira a migliorare le fasi di codifica e selezione, necessarie per trasformare i dati mutazionali disponibili in caratteristiche rilevanti per le tecniche di apprendimento automatico. Inoltre, si propone di ottimizzare la fase di predizione e i suoi risultati attraverso la valutazione e il miglioramento della pipeline precedentemente proposta basata sulla Data Science. L'obiettivo è identificare la migliore strategia in termini di prestazioni e mettere in evidenza le caratteristiche rilevanti preservate nei metodi proposti. Le caratteristiche rilevanti risultanti potrebbero essere quindi un punto di partenza per terapie personalizzate per i pazienti affetti da CRC che non rispondono alle terapie convenzionali.

Parole chiave: Famiglia del gene RAS, Machine Learning, mutazioni, selezione delle caratteristiche, bioinformatica, cancro colorettale, codifica delle caratteristiche, genomica computazionale.



Contents

Introduction

The genetic material, or genome, is given by the set of genes, which are portions of DNA that contain information necessary for the proper functioning of the organism. Variation in the structure of the genetic material, and so in the nucleotide sequence of DNA, is called mutation and can cause diseases such as cancer. Much mutational data is possible today thanks to the development in recent decades of NGS, a technology that allows the entire genome to be sequenced in a short time and at a low cost.

Within this context, this thesis is developed. Our main objective is the identification in RAS-mutated colorectal cancer (CRC) patients of co-occurrent mutations, whose actionability may be further investigated. Specifically, the study aims to assess and enhance a previous workflow[1], in which a Data Science-based pipeline has been proposed to identify the most frequently co-occurring mutated genes in a RAS-mutated subpopulation of CRC patients. In particular, this thesis is focused on improving the encoding and selection phases, needed to transform the available mutational data into relevant features for Machine Learning techniques. In addition, it evaluates and enhances the previously proposed Data Science-based pipeline to better optimize the prediction phase and its results, together with the strategy required to identify relevant co-occurrent mutations.

To achieve this goal, we work on several aspects. To begin with, for what concern the encoding phase, we develop a method for identifying the category of patients, called hypermutants, who can be eliminated from the analysis because they are subject to a specific therapy. Moreover, we perform a sensitivity analysis to estimate suitable significance thresholds for MutSig and MutClust algorithms, needed to identify a gene

space on which proceeding with the analysis and to determine any mutational area -hotspot- of interest.

The other aspects we focus on concerns the feature selection and prediction phases. We investigate the robustness of Lasso Logistic Regression selection for Logistic regression model using a bootstrapping approach to select the most conserved features. Furthermore, we evaluate an alternative selection based on feature importance for the classification task of interest (such as Mean Decrease in Impurity) and we also run different Machine Learning models within different scenarios to evaluate the performances in terms of different metrics (such as accuracy, precision, recall and f1-score) and the selected features spaces. In this way, using different techniques and analyses can help better determine features that are strongly preserved in the proposed methods and assess their robustness.

By performing all these analyses and evaluations, we highlight features that could offer a starting point for personalized therapies for CRC patients who do not respond to conventional therapies.

1 Background

In this chapter, the fundamental concepts that are the basis for understanding the mechanisms that act at the basis of biological systems are illustrated. In addition, an overview of colorectal cancer and Next Generation Sequencing (NGS) technology is given.

1.1. DNA, Genes, Mutations

DNA (desoxyribonucleic acid) is a macromolecule found in all cells of living beings. Its typical double-helix shape is the result of the double-helix coiling of two polynucleotide chains. These chains are chemically composed of nucleotides, elements formed by a phosphate group, a deoxyribose sugar and a nitrogenous base (adenine, cytosine, guanine or thymine). Nucleotides within the same chain are linked together by a covalent bond, whereas those between the two chains are linked by a hydrogen bond. The double helix is packed, with the presence of proteins called histones, until it takes on a compact, twisted shape (called a chromosome) that thus allows DNA to be contained within the nuclear membrane in eukaryotic cells. During cell division, the chromosome takes an “X” shape and each human cell contains 23 pairs: 22 equal pairs called "homologous," 1 pair called "sexual" because it determines male (XY) and female (XX) sex.

DNA plays the crucial role of containing and transmitting essential information for the proper functioning of the organism. Precisely, individual instructions are contained within genes. Genes, in fact, are portions of DNA that contain the information for coding a product (mainly proteins). Proteins, in their turn, are important to ensure the proper functioning, structure, and shape of the cells that make up organs and tissues.

In a nutshell, the protein-coding mechanism involves transcribing regions of DNA into RNA and then translating them into protein. During the transcription process, which takes place in the nucleus, DNA straightening, separation of the two chains, and generation of messenger RNA (mRNA) occur. Subsequently, mRNA leaves the nucleus and arrives in the ribosome, where the translation phase takes place. At this stage, a nucleotide triplet (called a codon) is translated into an amino acid. As a result, the amino acids conjunction forms the protein. Moreover, translation occurs according to the genetic code.

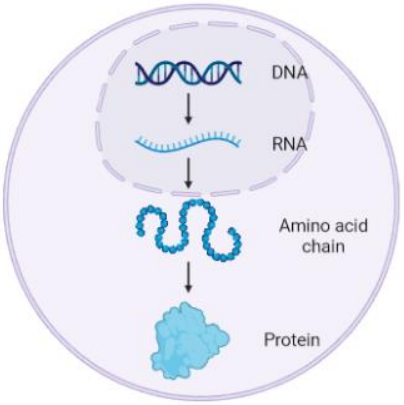


Figure 1. Example of gene expression in a cell

A recent study [2] reported a catalogue of human genes in which 41,356 genes were counted, 19,839 of which encode a protein (i.e., transcribed into RNA and then translated into protein). Taking the number of base pairs into account, merely 2% of DNA is coding. Lastly, the non-coding portion of DNA (called Junk DNA) plays additional key roles, such as regulating gene expression, i.e., whether and how much a gene is activated.

In addition to the process of DNA transcription and translation, the process of DNA duplication also takes place in cells. Specifically, this occurs during cell division, in which a mother cell divides into two or four daughter cells (depending on whether it is mitosis or meiosis). Briefly, during the stages of cell division, DNA is "unwound" from its double-helix form, the two strands are separated, and then complementary

strands are created. The result is double DNA destined for daughter cells. Again, there are several steps involving multiple elements making this process very complex and high risk of error.

Given the complexity of the mechanisms, the number of elements involved, and the possible occurrence of mutagenic factors (external factors, such as exposure to ionizing radiation), errors can occur. Nevertheless, errors can be recognized by the cell, which brings DNA repair mechanisms into play. Yet, if the error is not recognized, it can be transmitted to subsequent cells and, if it occurs in a germ cell, even inherited by every cell of an offspring organism. Mutations are the basis of evolution since they allow organisms to differentiate more, but they can also lead to the onset of diseases and/or dysfunctions of the organism.

To sum up, mutations can be classified according to:

1. Type of cell in which they occur.
2. Which DNA sections it involves.
3. Origin (induced or spontaneous).

1.1.1. Type of cell in which they occur

Whether a mutation first occurs in a somatic cell, it can be transmitted only to cells of the individual in which it occurs. On the other hand, if the mutation occurs in a germ cell (such as the egg cell or sperm cell), it can pass on to subsequent generations, causing mutated offspring.

1.1.2. Sections, types and effects

Depending on which DNA sections it involves, mutations can be divided into:

1. **Gene mutations**: these affect single nucleotides (point) or a few base pairs (for repeated sequences).

2. **Chromosomal mutations:** these affect an area of the chromosome.
3. **Genomic mutations:** these alter the number of chromosomes.

The most common gene mutations are the point mutations listed here below:

1. **Substitution**: substitution of one nucleotide/base pair.
2. **Insertion**: addition of one or more nucleotides/base pairs.
3. **Deletion**: loss of one or more nucleotides/base pairs.

The effect of these mutations is defined by categorizing mutations as follows:

1. **Missense**: it causes a change in coding for a different amino acid. The phenotype can change depending on the substituted amino acid.
2. **Nonsense**: causes a change in coding for an amino acid to a STOP codon, terminating protein synthesis.
3. **Silent**: if the pair substitution encodes an equivalent codon.
4. **Frameshift**: insertion and deletion alter the reading sequence causing a shift resulting in the insertion of different amino acids.

1.2. Next Generation Sequencing

Next Generation Sequencing (NGS) is a DNA sequencing technology that enables large genomes to be analysed in short time frames. These tools make it possible to identify many alterations, some of which may prove to be an important aid in therapeutic decision making.

The Next Generation Sequencing process consists of:

1. **Sample preparation**: DNA is extracted from the sample and fragmented. The fragments are ligated to adapters (artificial molecules) that enable their amplification.

2. **Amplification of DNA fragments**: various techniques such as PCR that allows multiple copies of the DNA fragments to be produced to be sequenced simultaneously.
3. **Sequencing**: technologies involve the generation of light or electrical signals.
4. Sequencing data are **processed** and analysed by special software that identifies any variations and mutations from the reference genome. The results are reported in BAM (Binary Alignment and Map), SAM (Sequence Alignment and Map) and VCF (Varian Call Format) files.

1.2.1. Applications

NGS technology can be used for several applications:

1. **Whole Genome Sequencing** (WGS): analysis of an individual's entire genome.
2. **Whole Exome Sequencing** (WES): analysis of the coding region of all genes in an individual.
3. **Targeted Sequencing**: analysis of a group of genes or a single gene.
4. **Transcriptome Analysis**: analysis of RNAs produced by the cell (transcriptome). [3]

These types of experiments allow different data-processing approaches to extrapolate information about mutational data, which can then be studied and analysed.

1.3. Colorectal cancer

Colorectal cancer (CRC) is the third most common cancer in industrialised countries. According to AIOM-AIRTUM 2021 data, it accounts for 10 % of all cancers diagnosed worldwide [4].

1.3.1. Anatomy

The digestive system consists of a set of organs responsible for the intake and digestion of food, absorption of nutrients and elimination of waste products. It includes: mouth, epiglottis, pharynx, oesophagus, liver, pancreas, small intestine, large intestine.

Secondly, the terminal tract, the large intestine, is responsible for absorbing water and electrolytes, accumulating food waste and providing for its decomposition and evacuation from the body. About 1.5 meters long in total, it consists of: Colon, Rectum and Anal Canal.

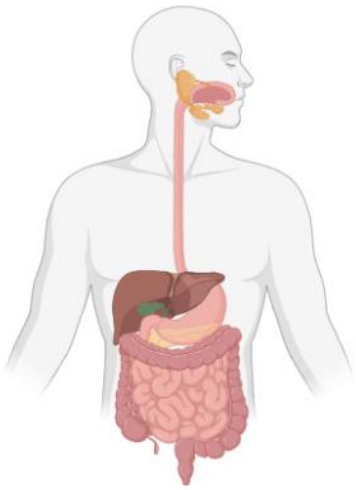


Figure 2. Digestive System

1.3.2. Classification

Colorectal cancer can be classified according to its histology, molecular mechanisms, molecular markers, and stages of disease.

1.3.2.1. Histopathological classification

Histologically (that is by the type of cells that constitute it), the most common colorectal cancer is adenocarcinoma (95%). Adenocarcinoma is a malignant tumour that develops from epithelial glandular cells that are assigned to produce mucus: they

proliferate uncontrollably and invade surrounding tissues. Furthermore, glandular epithelial cells are present in the colon and rectum, lining the inside of the colon and producing mucin.

1.3.2.2. Molecular classification

Based on the molecular mechanisms underlying disease formation and development, colorectal cancer can be classified into two broad categories:

1. 85% - MSS (stable microsatellite) or MSI-L (unstable low-level microsatellite) phenotype: among these, 1% represent hereditary FAP (familial adenomatous polyposis) syndrome and 84% sporadic.
2. 15% - MSI - H (high-level unstable microsatellite) phenotype due to DNA mismatch repair deficiency: 3% hereditary Lynch syndrome and 12% sporadic.

1.3.2.3. Classification based on molecular markers

The classification below is based on the use of specific molecular biomarkers to define the biological features of the tumour.

1. **Infiltration with immune system cells**: local immune cell infiltration has been shown to be a powerful factor for prognostic classification: The MSI-H phenotype is closely associated with high lymphocyte density, an association in all probability ascribable to a pronounced antitumor immune response.
2. **Microsatellite instability**: Analysis of microsatellite instability may provide information on the prognosis and therapeutic response of patients: MSI-H phenotype did not show benefit from adjuvant fluorouracil therapy whereas had an improved response to irinotecan-based chemotherapy.
3. **RAS gene family mutations**: Mutations in the RAS gene family (especially KRAS oncogene) do not allow affected cells to respond to treatment with anti-

EGFR antibodies, thus reducing response rates from monotherapy from almost 20% to nearly 0%.

1.3.2.4. Cancer stage classification

"Stage" is the term used to describe the size of the tumour and its possible spread. Accordingly, this information is used by physicians to determine the best therapy. Speaking of colorectal cancer, it is classified into:

1. **Stage I**: tumour circumscribed within the intestinal wall.
2. **Stage II**: intestinal patter invaded by the tumour but lymph nodes unharmed.
3. **Stage III**: one or more lymph nodes close to the intestine invaded by the tumour.
4. **Stage IV**: tumour spread to other organs. [5]

1.3.3. Therapeutic treatments

Therapeutic treatments depend on the area, staging, presence or absence of metastasis. Potential treatments are:

1. **Resective surgery**: removal of the tumour mass.
2. **Chemotherapy** (therapy directed at tumour cells) and/or immunotherapy (acts by stimulating immune response)
3. **Postoperative (adjuvant) chemotherapy**: based on the analysis performed on the surgically removed tumour, it is evaluated whether to perform chemotherapy therapy to reduce the risk of recurrence.
4. **Postoperative radiation therapy**: in support of patients with cancer tumour resection.

2 Thesis Goals

As mentioned in the introduction, this study is focused on colorectal cancer (CRC) patients with mutations in the RAS gene family (RAS, KRAS, NRAS): these patients are, in fact, unlikely to respond to conventional therapies and mutations on genes of the RAS gene family are not actionable (i.e. valid targets for alternative treatments). Accordingly, our main objective is the identification in RAS-mutated CRC patients of co-occurrent mutations, whose actionability may be further investigated.

Specifically, the study aims to assess and enhance a previous workflow developed within a master thesis titled "*Statistical and machine learning methods for discovering mutational signatures in RAS-mutated colorectal cancer patients*" [1]. In such a research work, a Data Science-based pipeline has been proposed to identify the most frequently co-occurring mutated genes in a RAS-mutated subpopulation of CRC patients. Particularly, a supervised learning setting is used to extract the features that are crucial to recognize RAS-mutated patients, starting such a prediction task from an entire set mutational features obtained using a combination of MutSig2CV [6], [7] and MutClustSW [8] algorithms.

Given the promising results of that work, this thesis project is first focused on improving the encoding and selection phases, needed to transform the available mutational data into relevant features for Machine Learning techniques. In addition, it evaluates and enhances the previously proposed Data Science-based pipeline to better optimize the prediction phase and its results, together with the strategy required to identify relevant RAS co-occurrent mutations.

Towards these goals, for what concerns the encoding phase we aim to develop a method for identifying the category of patients, called hypermutants, to whom specific

therapy is administered, would help in achieving the goal as these patients would thus be eliminated from the analysis.

Moreover, we aim to better determine genes and hotspots of interest for the application of machine learning models through investigations and a sensitivity analysis able to tune the significance thresholds of MutSig2CV and MutClustSW. Accordingly, we aim to subdivide MutSig2CV-based assessments into subcases (such as genes that belong only to patients with the mutated RAS gene family, genes that belong only to patients with the unmutated RAS gene family, genes that are in common between the two above categories) to better identify a gene space of interest to proceed with MutClustSW analysis.

For what concerns the feature selection and prediction phases, we aim to evaluate several scenarios, feature selection methods and classification models. Using different methodologies in different training/testing scenarios evaluated in terms of performance with different metrics (such as accuracy, precision, recall, and f1-score) offers the possibility both to choose the best strategy in terms of performance and extract the most relevant features. Specifically, we aim to analyse the robustness of a Lasso Logistic Regression classifier to accurately choose the most conserved features and make comparisons with other classification models. Also, we aim to apply an alternative selection method based on feature importance to find other spaces that could be valuable in terms of the classification task of interest. Notably, we aim to compare so-obtained feature sets and produce rankings able to evaluate the predictive role of any selected feature.

The expected result is to highlight several mutational features that could offer a relevant starting point for further studies moving towards personalized therapies for those CRC patients who do not respond to conventional therapies.

3 Material

This section reports how the data collections are performed and how mutational data are processed to obtain final dataset.

3.1. cBioPortal

With NGS technologies that made it possible to investigate the genome more easily, genomic data has increased exponentially. Several projects have collected and stored genomic data in repositories and databases accessible to researchers to share and boost advances and discoveries in the genomic field within the scientific community.

The database from which the data considered in this study are taken is cBioPortal [9], [10]. cBioPortal allows interactive exploration of cancer genomics datasets and organises information into MAF files (Mutation Annotation Format), a standard file type aggregating mutational information from files in VCF format (Variant Calling Format reports somatic variants detected by variant identifiers).

To be specific, MAF files can be of two types:

1. **Minimum MAF** with 6 required columns:
   - Chromosome: affected chromosome (e.g. ch1).
   - Start_Position: lowest numeric position of the reported variant on the genomic reference sequence/the mutation start coordinate.
   - End_Position: highest numeric genomic position of the reported variant on the genomic reference sequence/the mutation end coordinate.
   - Reference_Allele: plus strand reference allele at this position; it includes the deleted sequence for a deletion or "-" for an insertion.

- Tumor_Seq_Allele2: primary data genotype for tumor sequencing allele.
- Tumor_Sample_Barcode: aliquot barcode for the tumor sample.

4 optional colums:

- t_alt_count: variant allele count (tumor).
- t_ref_count: reference allele count (tumor)
- Protein_position: relative position of affected amino acid in protein.
- SWISSPROT: UniProtKB/Swiss-Prot accession.

2. **Extended MAF**: 32 columns, including the previously mentioned, 1 column with amino acid variation, 4 columns with information on the number of reference alleles and variants in tumor and normal samples.

In addition to the mutational information (such as gene symbol, position in the chromosome, and variant type), clinical data are also available with information regarding patients (such as sex, age diagnosis, tumor site) and samples (such as tumor stage). The final dataset of each study is obtained by aggregating the information from the mutational and clinical dataset.

3.2. Datasets

In this thesis work, the datasets collected concern somatic mutations of colorectal cancer samples processed through whole-exome sequencing (WES) and using hg19/GRCh37 variant as the reference genome call.

The studies considered are:

1. *Giannakis et al*. [11]
2. *TCGA - Pan Cancer Atlas* (A project funded by the US National Institutes of Health (NIH), aimed at creating a catalog of genetic mutations responsible for cancer).

3. *Seshagiri et al*. [12]

Below is a summary table with key information from each study:

	Giannakis et al.	TCGA Pan Cancer Atlas	*Seshagiri et al.*
Patients	619	594 (*)	74 (*)
Samples	619	528 (*)	72 (*)
Patients with the mutated RAS gene	215	258	39
Percentage of patients with the mutated RAS gene compared with total patients	43,4 %	34,7 %	52,7 %

Table 1. Summary information about the collected data

(*) The number of patients and samples is different becausenonly samples subjected to WES are considered.

3.2.1. Protocols

Regarding the protocols implemented by each study, the flow consists of the following general steps:

1. **Sample preparation and extraction**: acquisition of genomic DNA from biological samples.
2. **Exon library preparation**: DNA fragmentation, ligation to adapter, target enrichment.
3. **Exome sequencing**: library subjected to parallel sequencing to produce millions of short reads.
4. **Alignment and mapping**: alignment of sequencing data to human genome reference sequence.
5. **Variant Calling**: in silico tools used to determine variant calling.
6. **Annotation**: annotation of variants provides information for analysis and interpretation.
7. **Filtering**: to identify random genes.

The protocols of each study are given below.

Giannakis et al. **Protocol details**	
Sample preparation and extraction:	Patients undergo resection of the tumor and adjacent tissue. Samples are fixed in formalin and embedded in kerosene (FFPE). Genomic DNA is extracted from dissected tumor areas from tissue sections obtained from FFPE blocks using QIAGEN QIAamp DNA FFPE Tissue Kit.
Exon library preparation	DNA subjected to hybrid capture in solution phase with Exome Sure Select V2 (Agilent Technologies).
Exome sequencing	Sequencing with Illumina HiSeq 2000 (average coverage 90x).
Alignment and mapping	Alignment Using Burrows-Wheeler Aligner BWA-MEM.
Variant Calling	Somatic mutation detection by MuTect and Somatic Indels with Idelocator and Strlka.
Annotation	Clinical, epidemiological and pathological annotations related to incident colorectal cancers in the Nurses' Health Study (NHS) and Health Professionals Follow-up Study (HPFS) cohort studies.
Filtering	Filter out consistent C>T mutations with a single-stranded bias based on read pair orientation.

Table 2. *Giannakis et al.* protocol details

TCGA Pan Cancer Atlas protocol details	
Sample preparation and extraction:	Biological samples were collected from patients who underwent surgical resection and had not received any previous treatment for their disease, including chemotherapy or radiation therapy. Each frozen tumor sample had an accompanying normal tissue sample. Each tumor and the adjacent normal tissue sample were embedded in optimal cutting temperature (OCT) medium. DNA extraction from the tumor samples using the Qiagen AllPrep DNA/RNA kit (Qiagen).
Exon library preparation	Exome capture was performed using SOLiD (NimbleGen CCDS Solution Probes) and Illumina (NimbleGen SeqCap EZ Exome 2.0 Solution Probes).
Exome sequencing	Sequencing with Illumina HiSeq 2000 (mean coverage 179x).
Alignment and mapping	Alignment using Burrows-Wheeler Aligner BWA - MEN (if read length is greater than or equal to 70 bp) or BWA - aln. Each read group is aligned to the reference genome separately, and all read group alignments that belong to a single aliquot are merged using Picard Tools, SortSam, and MergeSamFile. Duplicate reads are marked to avoid downstream variant calling errors.
Variant Calling	Variant Calling is performed using four separate pipelines-Musa, MuTect2, VarScan2, Pindel. Variant Calling is reported by each pipeline in a VCF format. The four separate pipelines are implemented to harmonise the data. Currently there is no scientific consensus on the best variant calling pipeline, so the researcher is responsible for choosing the most appropriate pipeline for the data.
Annotation	Raw VFC files are annotated with the VEP (Variant Effect Predictor) v84 command. Variants in VCF files are also matched with known variants from external mutation databases: GENCODE v22, Sift v.5.2.2, ESP v.20141103, Polyphene v.2.2.2, dbSNP v.156, Ensembl genebuild v.2017-07, Ensembl regbuild v.13.0, HGMD public v.20154, ClinVar v.201601
Filtering	False-positive filter for labeling low-quality variants in VarScan.

Table 3. TCGA Pan Cancer Atlas protocol details

Seshagiri et al. protocol details	
Sample preparation and extraction:	Fresh frozen primary colon tumors matched to the patient and normal tissue samples were obtained from commercial sources. Tumor DNA extracted using the Qiagen AllPrep DNA/RNA kit (Qiagen).
Exon library preparation	Exome capture was performed using SeqCap EZ v2.0 (NimbleGen).
Exome sequencing	Sequencing with Illumina HiSeq 2000 (average coverage 179x).
Alignment and mapping	Alignment Using Burrows-Wheeler Aligner BWA.
Variant Calling	Detection of somatic mutations with Illumina 2.5M single-nucleotidepolymorphism (SNP) and Indels with GATK indel Genotype Version 2.
Annotation	Variants annotated using Ensembl (version 59).
Filtering	Known germline variations represented in Known germline variations represented in dbSNP Build 131, but not represented in COSMIC v54.Variants that were present in both tumor and normal samples were removed as germline variations.Predicted somatic variations were further filtered to include only those positions with a minimum 10-fold coverage in both tumor and corresponding normal.

Table 4. *Seshagiri et al.* protocol details

3.2.2. Deletion of duplicated rows in TCGA Pan Cancer Atlas

In examining the columns of the TCGA Pan Cancer Atlas dataset, special regard was placed on the "Matched_Norm_Sample_Barcode" column, in which the patient ID is matched to the normal sample. For the same patient, two rows resulted with the same mutational information minus the code matched to the normal sample, i.e., blood (10) or tissue (11).

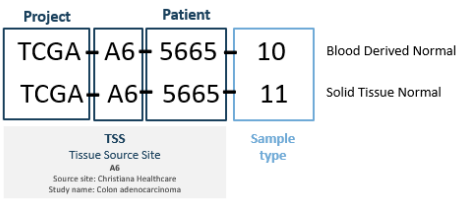


Figure 3. Example of two rows in the Matched Norm Sample Barcode column of the same patient with two different codes to indicate normal sample type.

Since two rows containing the same mutational information minus the code matched to the normal sample resulted for the same patient, specifically for 37 patients, we proceeded with the elimination of one of the two duplicate rows: specifically, the one with blood as sample type (10) is retained to be consistent with other patients in the dataset. In addition to maintaining consistency, computational performance is also improved by reducing the number of rows.

4 Methods and Software

This part explains in detail how the models and software used in this analysis work. Furthermore, we introduce the metrics used to quantify the level of performance of the models discussed in the next chapter.

4.1. MutSig2CV

MutSig (Mutational Significance)[6], [7] is an algorithm implemented in Matlab to recognise genes mutated with a higher frequency than expected by chance in a cohort of patients. In addition, MutSig developers collected samples from 21 cancer types by compiling a list of 18 388 genes.

The latest "MutSig2CV" version of MutSig algorithm estimates the background mutation rate (BMR, average frequency at which genetic mutations occur in a population) for each gene-patient-mutation category combination based on the silent mutations observed in the gene and non-coding mutations in the surrounding regions. Three significance tests are calculated for each gene:

1. **MutSigCV** (Covariance): determines the P-value for observing the given amount of non-silent mutations in the gene, given the background pattern determined by silent (and noncoding) mutations in the same gene and neighbouring genes in the covariate space.
2. **MutSigCL** (Clustering): groups mutations at the local site level, which allows MutSig to differentiate between, on the one hand, genes with uniformly distributed mutations and, on the other hand, genes with localised hotspots, assigning higher significance to the latter.

3. **MutSigFN** (Conservation): estimates the significance of the tendency of mutations to occur at highly evolutionarily conserved positions (using conservation as a proxy for likely functional impact).

Finally, these three statistical tests are combined into a single P-value. At first, an earlier joint P-value (CL + FN) is calculated from the joint probability distribution of random permutations. Afterwards, this is combined with the MutSigCV P-value using the Fisher method (which combine P-values from independent tests).

Concerning the high number of genes, the final values of P-values are converted to False Discovery Rate (FDR), taking the name q-value, using the Benjamin Hochberg method. The Benjamin Hochberg procedure enables the selection of significant values in a set of independent statistical tests. Briefly, a threshold value is established for the rate of acceptable false positives (for example 5%) and the P-values obtained from the independent tests are sorted in ascending order. Subsequently, for each P-value, the ratio of acceptable false positives to the position of the P-value in the sorted list is calculated. Genes with a q-value <0.1 are declared significantly mutated.

4.2. MutClustSW

MutClustSW [8] is an algorithm that identifies somatic mutations that occur most frequently in a genomic region (called hotspot). Specifically, it is based on the Smith-Waterman algorithm to identify mutation hotspots of single or clustered amino acid residues, as mutation hotspots are single or clustered amino acid residues that show a high mutation frequency in cancer-related genes. The MutClustSW algorithm evaluates the clustering of mutation hotspots and their position in individual genes: in conclusion, it can identify both single amino acid mutation hotspots and amino acid tracts with high mutation frequencies without a priori information on protein domains.

An adaptation of the Smith-Waterman algorithm (originally used local alignment of DNA and protein sequences) is made: mutations along the amino acid length of given genes are converted into one-dimensional series or score vectors representing the presence or absence of mutations. If the amino acid is not mutated, then a negative score is assigned at that position. On the contrary, the score increases with the frequency of mutation of that amino acid at that position. Hotspots are identified by recursively iterating the algorithm: at each iteration, the nearest hotspots below a certain threshold are merged into a single cluster.

Significance values are then adjusted for multiple testing using the Benjamin-Hochberg method, and segments with a false discovery rate (FDR) < 0.05 are defined as significant hotspots. MutClustSW is an open-source algorithm implemented in R, available for free download and use.

4.3. Machine Learning Classifier

Machine Learning is a branch of Artificial Intelligence concerned with developing algorithms that enable computers to learn tasks and improve automatically by analysing large amounts of data. Supervised Machine Learning models try to find patterns and relationships among the example data to predict the outcome of a learnt task in the case of new data.

Broadly speaking, the machine learning workflow consists of the following steps:

1. Collection and preprocessing of data
2. Choice of the algorithm(s)
3. Pattern analysis and parameter selection
4. Training of the model
5. Testing of the tuned and trained model on new data
6. Interpretation of the collected results

Specifically, Supervised Machine Learning approaches aim to find a relationship between dependent variables and independent variables in known training data (used as examples) in order to make predictions on new data. The dependent variables are those that are to be predicted (e.g., memberships, regession values or class labels), while the independent variables allow to estimate these predicted variables (e.g., classify an observation into a particular class).

All Machine Learning models presented in this study are implemented in Python for a classification task. Details of the Supervised Learning methods are given below.

4.3.1. Logistic regression, Lasso and Ridge regularisations

In comparison to linear regression, which looks for a line that best fits the data, logistic regression is used to model the probability of a finite number of outcomes (e.g., class membership in a classification task).

Lasso and Ridge regularisations are embedded feature selection methods since they both attempt to minimise the sum of residuals (RSS) along with a penalty term. In the first case (L1 norm - the sum of the absolute values – and the penalty term defined as L1), a shrinkage of the variables is performed as it can reset some coefficients of the model to zero producing a simpler model. In the second case (L2 norm - square root of the sum of the squared values - and penalty term defined as L2), on the contrary, the variables are all retained, without eliminating them but reducing their impact on the model.

In Python we used the *sklearn.linear_model*, specifically implementing with the class *LogisticRegression*.

4.3.2. Random Forest and Mean Decrease Impurity (MDI)

Random Forest is a machine learning model that is based on the creation of multiple decision trees. Each tree is created using a random subset of training data and a subset

of independent variables. Therefore, combining the results of all the individual trees and choosing as the classification the one with more outputs produces a final prediction. In Python, we used the *sklearn.ensemble* library implemented with the class *RandomForestClassifier*. The Random Forest with Feature Importance (called Mean Decrease impurity) calculates the importance of each feature as the sum of the number of splits (across all trees) that include the feature in proportion to the number of samples it splits.

In Python, we used the *sklearn.ensemble* implementation of the class *RandomForestClassifier*.

4.3.3. Bootstrapping

Bootstrapping, which consists of repeated samplings of the original dataset with replacement, gives an estimate of the sample distribution.

In this thesis, the Bootstrapping method has a dual role. The first role consists in feature selection: in particular, after applying 100 Lasso Logistic Regression models, we decided to select features that are different from zero at least 50 % of the times. The features selected by this method are then compared with those selected by the Random Forest method with Feature Importance to assess which and how many turn out to be in common. The second one is to evaluate the robustness of the individual Lasso Logistic Regression model: specifically, the features are ranked in descending order according to the number of times they are selected in the Bootstrapping model. In this way, it is possible to evaluate the features that are most often selected in Bootstrapping so that they can be compared with those selected by the Lasso Logistic Regression model.

In Python we used the class *LogisticRegression* of the *sklearn.linear_model* and the *sklearn.utilis* library with its class *resample* to obtain 100 boostrapped versions of Lasso regularized Logistic Regression models.

4.3.4. Interpretation of results: SHAP

After applying Machine Learning models, it is crucial to use methods intended to enable users (like researchers or clinicians) to interpret the models and make decisions based on their predictions. Two different types of analysis can be performed:

1. Global-level analysis determines the most important features by analysing the predictions made for the entire dataset.
2. Local-level analysis determines, for each prediction, which variables were the most important given the specific instance.

One such method is the SHAP algorithm, implemented in a Python library named SHAP (Shapley Additive exPlanations) and considered in this study. The SHAP algorithm originated in the context of game theory to determine how much each player contributed to the success of a collaborative game. In Machine Learning, SHAP evaluates the difference between the model prediction for the entire dataset and the model prediction for each input instance, taking into account the various combinations of features that may affect the model prediction. Thus, a better interpretation of the contribution of each feature to model prediction is obtained. In this study, the features are plotted in a bar graph in which the average absolute value of each feature over all input instances used to train the Machine Learning model is represented.

4.4. Metrics

Several metrics are analysed to evaluate the performance of the proposed models. Usually, machine learning binary classification models take into account the confusion matrix, which is a 2x2 table summarising the number of correct and incorrect predictions made by the model.

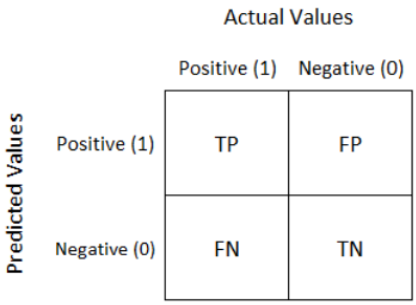


Figure 4. Example of a two-class confusion matrix

From such a confusion matrix, typically, we can evaluate the following metrics:

1. **Accuracy**: assesses the goodness of a classification model. It is defined as the ratio of the number of correct predictions to the total number of predictions made by the model.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. **Precision**: measures the model's ability to avoid misclassifying negative samples as positive. It is defined as the ratio of the number of true positives to the total number of cases predicted as positive by the model.

$$Precision = \frac{TP}{TP + FP}$$

3. **Recall**: measures the ability of the model to identify all positive samples. It is defined as the ratio of the number of true positives to the total number of truly positive cases in the dataset.

$$Recall = \frac{TP}{TP + FN}$$

4. **F1-score**: assesses the model's ability to balance precision and recall. It is defined as the harmonic mean between precision and recall. 2 * (precision * recall) / (precision and recall). It is a value between 0 and 1 where a value close to 0 indicates that the model has no prediction capability, in contrast, a value close to 1 indicates perfect accuracy between precision and recall.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

In Python is used the *sklearn.metric* library implemented with the class *classification_report.*

5 Results and Discussion

This chapter explains the several steps implemented in this study. Initially, data processing is carried out, including identification and subsequent elimination of hypermuted patients, merging the datasets and creating three scenarios. Subsequently, relevant genes are selected using MutSig2CV, hotspots are identified using MutClustSW, and the occurrence matrix is created. Finally, a Supervised Machine Learning approach is taken to extract the mutational signatures. In addition, the results obtained in each step are applied and compared for all three scenarios described above.

5.1. Data Processing

5.1.1. Deletion of hyper-mutated patients

In MSI-H (high microsatellite instability) colorectal cancers (15%), cancer cells have higher frequency of mutation and repair processes of small, repeated DNA sequences. Thus, several genetic alterations are accumulated that are easily recognised by the immune system such that the immunotherapy approach to this type of tumour is effective by administering checkpoint inhibitor drugs to the patient. Patients with this characteristic are termed hypermutated and are eliminated from further analysis, since they already have immunotherapy as a valid treatment option.

Mutational assessment of the tumour allowed us to identify the hypermutated patients, most likely to respond to immunotherapy. One parameter used for mutational assessment is the mutation rate (or mutation rate), which indicates the rate at which mutations occur.

In this analysis, the following formula is used to calculate the mutation rate:

$$\textbf{Mutation rate} = \frac{\textbf{Sum total mutation length } (\textit{for each patient independently of genes})}{\textbf{Total sum of the length of the genes in the dataset}}$$

Calculating the numerator is simple since the mutation length information is obtained from the VCF file and therefore already present in the dataset. As for, however, the denominator the calculation of the length of the genes present in the dataset is not trivial. We have implemented this aspect using the class "*EnsembleRelease*" from "*pyensambe*" library Python through which, using Hugo Symbol of the gene, it is possible to derive its respective length in amino acid bases.

The following graphs (in logarithmic scale) are obtained for the three studies:

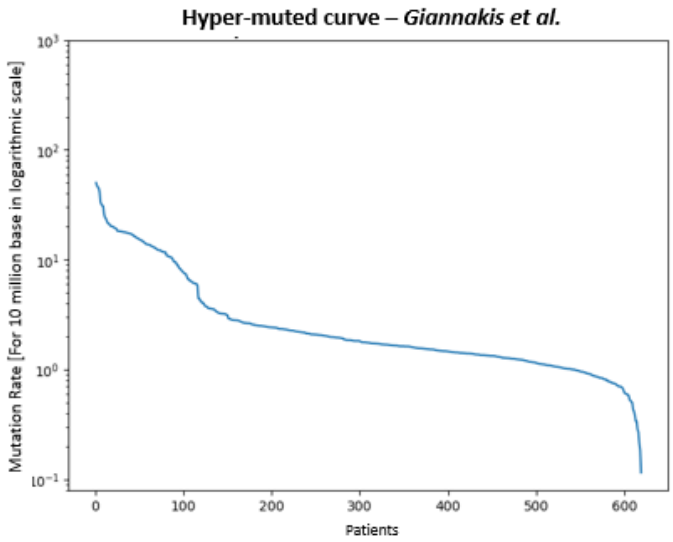


Figure 5. Curve showing on the x-axis the patients and on the y-axis the associated mutation rate per 10 million bases on a logarithmic scale from the study by *Giannakis et al.*

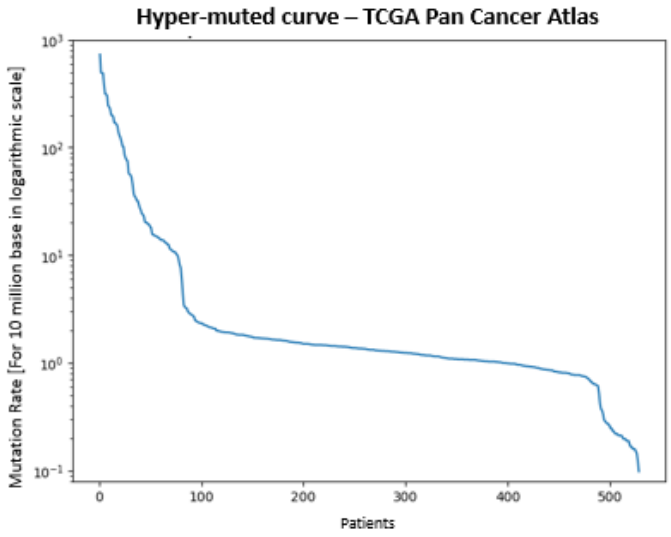


Figure 6. Curve showing on the x-axis the patients and on the y-axis the associated mutation rate per 10 million bases on a logarithmic scale from the study by *TCGA Pan Cancer Atlas.*

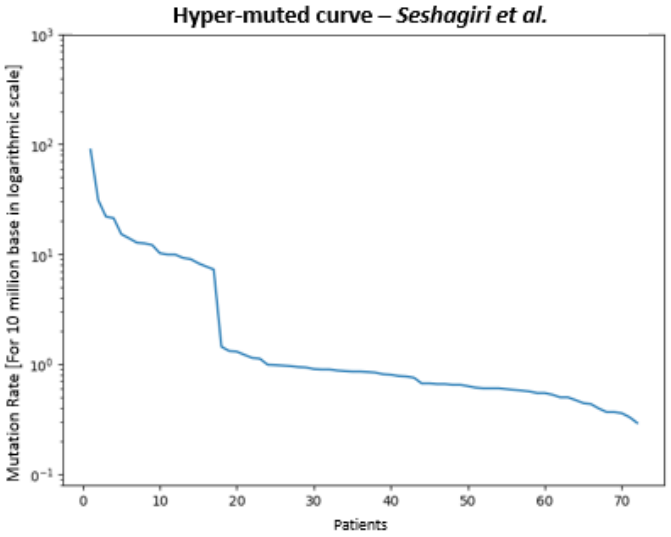


Figure 7. Curve showing on the x-axis the patients and on the y-axis the associated mutation rate per 10 million bases on a logarithmic scale from the study by *Seshagiri et al.*

The three graphs show a similar trend: there is a large slope at first, then a plateau and finally another slope. The point of interest that separates the hyper-muted patients from the non-hyper-muted is the one that follows the first slope and begins the plateau phase.

To identify this point (i.e., the threshold that corresponds to the number of hyper-muted patients) the following approach is proposed and developed.

The method uses the *KneeLocator* python function of the *kneed* library that locates the knee/elbow points of a line fit to the data. The knee/elbow is defined as the point on the line with the maximum curvature. The parameters set are curve = "convex" and direction = "decreasing," since an elbow (convex) with negative slope (starting from the left, decreasing) is sought.

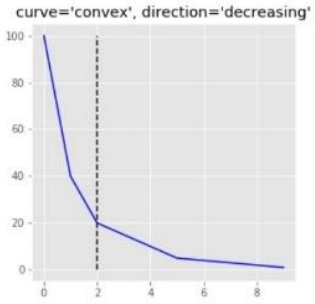


Figure 8. Example of application of the KneeLocator function using the "convex" and "decreasing" parameters.

As in the example in Figure **8** the KneeLocator function determines the point to the right of the slope (starting from the left).

The KneeLocator function identifies only one point. To obtain the one that identifies the threshold of hyper-mutated patients, we proceed iteratively by decreasing the dimensionalities of the patients and mutation rate each time an elbow is found (by putting the point preceding the newly found elbow as the new extreme). In this way, elbows are searched on all patients. For each point found (x), we consider the mutation rate of the tenth previous (x-10) and next (x+10) patients and calculate the difference in mutation rates between the patient found by the function.

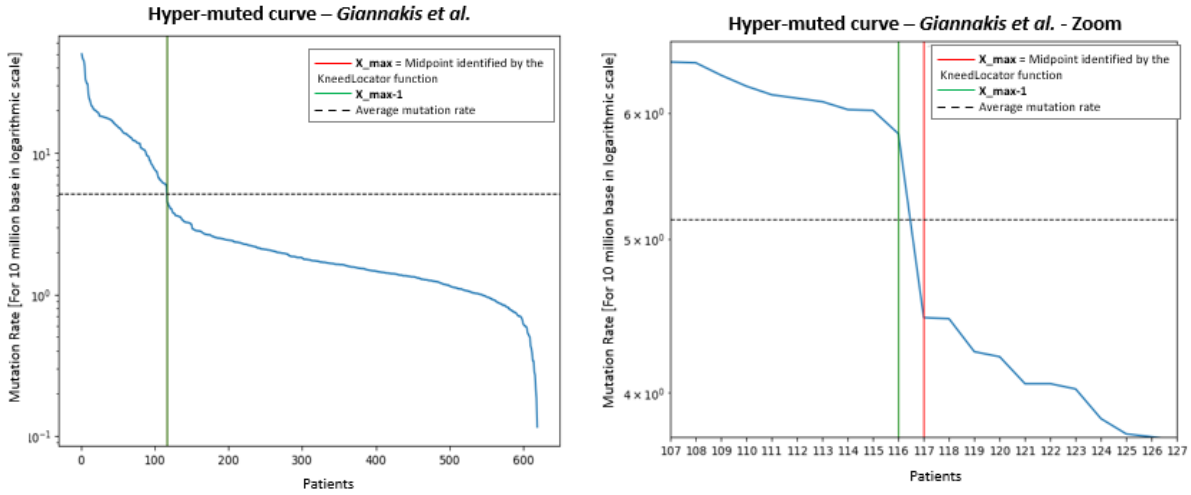
5.1.1.1. Example of application

1. Consider the entire dataset; (*Seshagiri et al*.:72 patients and 72 corresponding mutation rates).
2. Apply the KneeLocator function (the first point identified by KneeLocator corresponds to the 68th patient) and consider the associated mutation rate.
3. Consider the previous patient with the associated mutation rate (i.e., patient 67-sixth).
4. Calculate the difference between the mutation rates (gap patient 67-68).
5. Define the newly calculated gap as gap_max and the point obtained from the KneeLocator function as x_max.(i.e., patient 68th).
6. Set the new extremes to reapply the KneeLocator function: from the first to the x_max -1 patient (i.e., patient 67th).
7. Proceed by performing step 2, 3 and 4.
8. If the newly calculated gap is greater than gap_max, replace gap_max with the new gap and x_max with the new point.
9. Set the new extremes and proceed until the function finds no more points.

Finally, taking the gap_max, we derive the mutation rates associated with the points x_max and x_max-1.

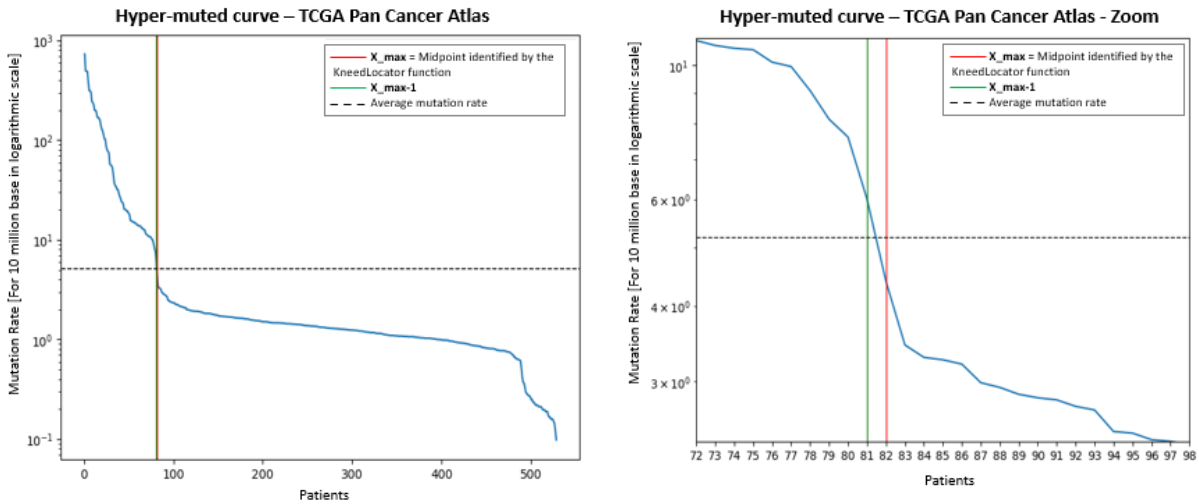
5.1.1.2. Results

The following results and graphs are obtained for the three studies:



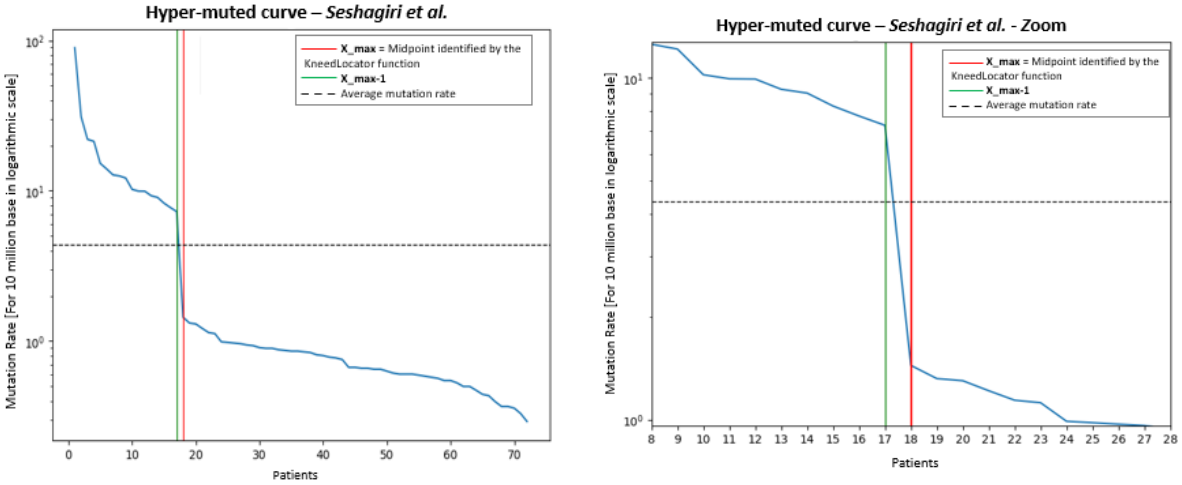
Giannakis et al.		
Patients	116	117
Mutation Rate	5,82	4,46

Figure 9. Graphs showing the results of the method for *Giannakis et al.*: red line (x_max), green line (x_max-1) and the black dashed line (calculated average mutation rate). The graph on the right shows a zoom of the affected area. The table shows the pair of patients identified by the KneeLocator function and the associated mutation rate.



TCGA Pan Cancer Atlas		
Patients	81	82
Mutation Rate	6,02	4,4

Figure 10. Graphs showing the results of the method for *TCGA Pan Cancer Atlas:* red line (x_max), green line (x_max-1) and the black dashed line (calculated average mutation rate). The graph on the right shows a zoom of the affected area. The table shows the pair of patients identified by the KneeLocator function and the associated mutation rate.



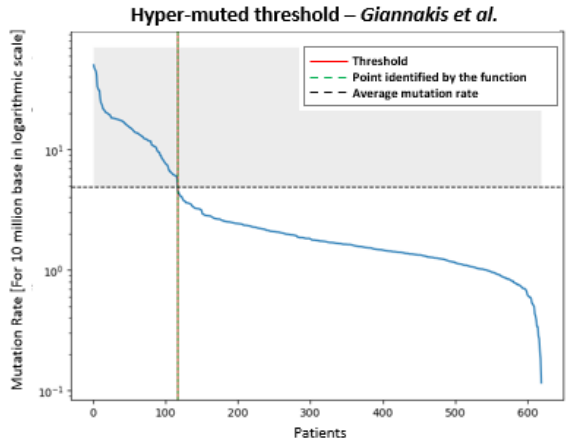
Seshagiri et al.		
Patients	17	18
Mutation Rate	7,27	1,44

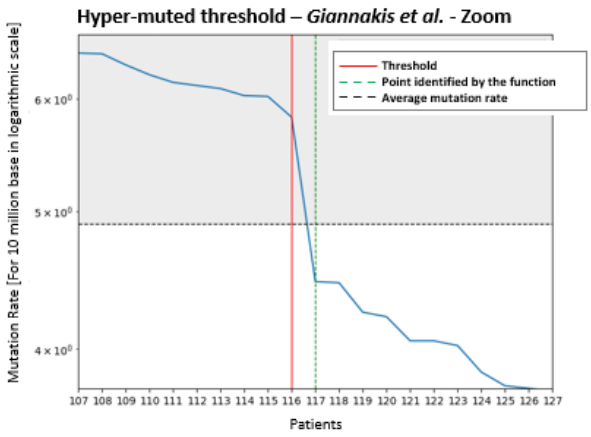
Figure 11. Graphs showing the results of the method for *Seshagiri et al:* red line (x_max), green line (x_max-1) and the black dashed line (calculated average mutation rate). The graph on the right shows a zoom of the affected area. The table shows the pair of patients identified by the KneeLocator function and the associated mutation rate.

The most stringent interval, i.e., the intersection of the three studies, is then considered. In our analysis, it is coincident with the interval of *Giannakis et al*.

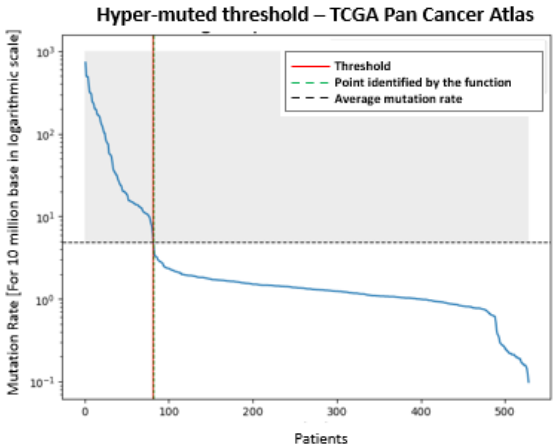
To determine a common threshold for the three studies, one considers the mean value of the most stringent interval and takes its integer value. In this case, the mutation rate for 10 million bases is equal to 5.

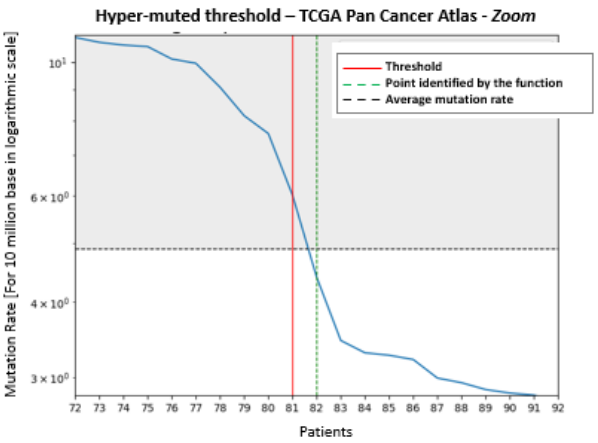
Then, the pair of patients with mutation rates surrounding the selected threshold are identified for each study. The patient of the pair whose mutation rate is higher than the threshold is considered as the cutoff.



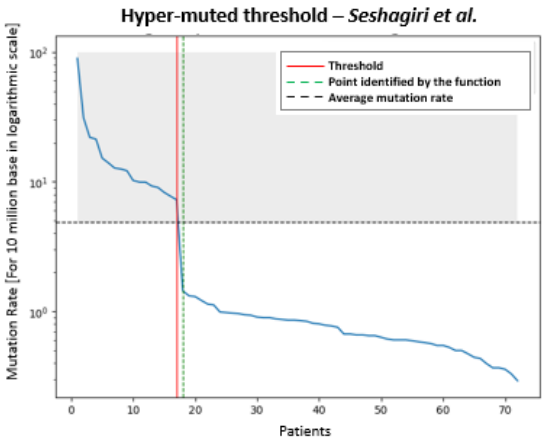


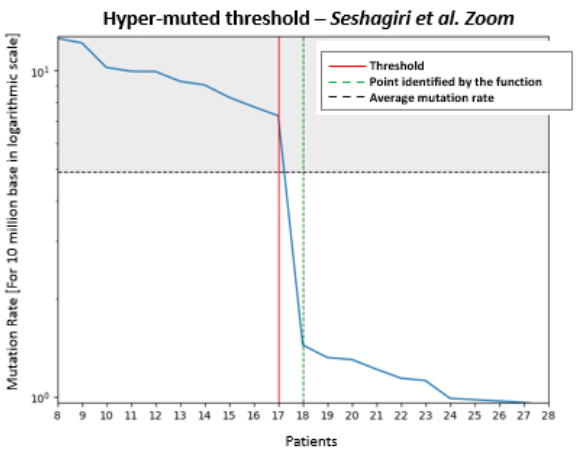
Giannakis et al.		
Patients	116	117
Mutation Rate	5,82	4,46





TCGA Pan Cancer Atlas		
Patients	81	82
Mutation Rate	6,02	4,4





Seshagiri et al.		
Patients	17	18
Mutation Rate	7,27	1,44

Figure 12. Results of the three datasets

The sum of hypermutated patients from the individual studies is **214** patients.

5.1.2. Merging datasets

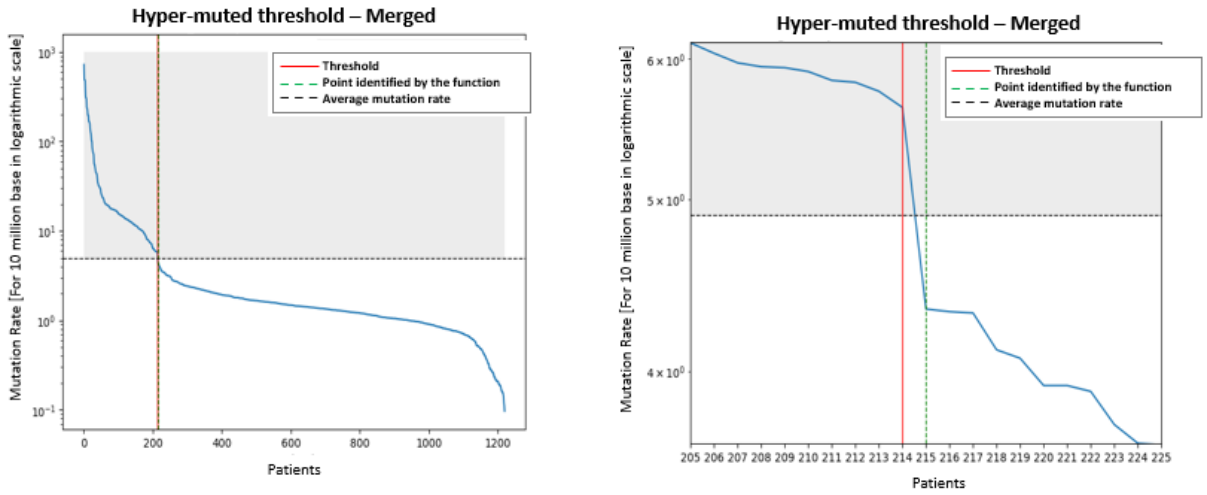
Even though the clinical data are heterogeneous with each other (not containing the same information, such as survival information), the mutational data are homogeneous with each other due to the fact that they are collected in MAF format from WES experiments, and thus the data can be merged with each other.

Considering the protocols implemented by each study (see 3.2.1 section), the merging of the datasets is done after the elimination of hypermutated patients. Below is the information on the merged dataset.

Merged Dataset without hyper-mutated patients	
Patients	**1005**
Samples	**1005**
Patients with the mutated RAS gene	**437**
Percentage of patients with the mutated RAS gene compared with total patients	**43 %**

Table 5. Summary information about merged dataset

The average mutation rate is also studied within a dataset composed of the union of the three studies with all patients, obtaining the following results:



Merged dataset		
Patients	214	215
Mutation Rate	5,63	4,33

Figure 13. Results of the merged dataset

From the results obtained, it is observed that the sum of hypermutated patients selected from the individual studies coincides with the number and IDs of hypermutated patients selected, using the same mutation rate value, from the merged dataset.

5.1.3. Subdivision of training and test datasets

After combining the data from the three studies, the merged dataset is divided into training and test dataset into three case scenarios:

1. 75 % training dataset and 25 % test dataset.
2. 80 % training dataset and 20 % test dataset.
3. 85 % training dataset and 15 % test dataset.

Each scenario has within its training and testing sets the same percentages of patients with mutated RAS gene and patients with non-mutated RAS gene as the total dataset (patients with the mutated RAS gene family are the 40 % of the total dataset and patients with the unmutated RAS gene family are the 60 %).

The following tables show the details of each scenario:

	75 % Training	**25 % Test**
Total Patients	755	250
Total patients with mutated RAS gene:	328	109
Total patients with mutated RAS gene from the *Giannakis et al*. study	137	46
Total patients with mutated RAS gene from the TCGA Pan Cancer Atlas study	165	55
Total patients with mutated RAS gene from the *Seshagiri et al.* study	26	8

	80 % Training	**20 % Test**
Total Patients	805	200
Total patients with mutated RAS gene:	349	88
Total patients with mutated RAS gene from the *Giannakis et al.* study	146	37
Total patients with mutated RAS gene from the TCGA Pan Cancer Atlas study	176	44

Total patients with mutated RAS gene from *the Seshagiri et al.* study	27	7

	85 % Training	**15 % Test**
Total Patients	854	151
Total patients with mutated RAS gene:	371	66
Total patients with mutated RAS gene from the *Giannakis et al.* study	155	28
Total patients with mutated RAS gene from the TCGA Pan Cancer Atlas study	187	33
Total patients with mutated RAS gene from *the Seshagiri et al.* study	29	5

Table 6. Details of the three scenarios

It is to be noticed that the patients in common among the three training datasets are 509 (including 193 patients with the mutated RAS gene), while among the three test datasets the patients in common are only 9 (including 2 patients with the mutated RAS gene).

5.2. MutSig2CV

After data collection and creation of the three scenarios, each training dataset is divided into training data containing patients with the mutated RAS gene and data of patients with the unmutated RAS gene. Both are given alternatively as input to MutSig2CV, which returns the file "*sig_genes.txt*" containing the list of significantly mutated genes with the corresponding p-value and q-value. Based on the values of the latter, the most relevant ones (those least likely to be mutated by chance) are selected. In this study, the following were highlighted:

1. Significantly relevant genes exclusive to patients with the mutated RAS gene.
2. Significantly relevant genes exclusive of patients with the unmutated RAS gene.

3. Significantly relevant genes given by the sum of the previous two cases.
4. Significantly relevant genes in common between patients with the mutated and non-mutated RAS gene.

5.2.1. MutSig2CV: Results and choice of gene features

Below are graphs showing the p-value and q-value (i.e., p-value converted to False Rate Discovery) on the x-axis and the number of genes on the y-axis for the three scenarios.

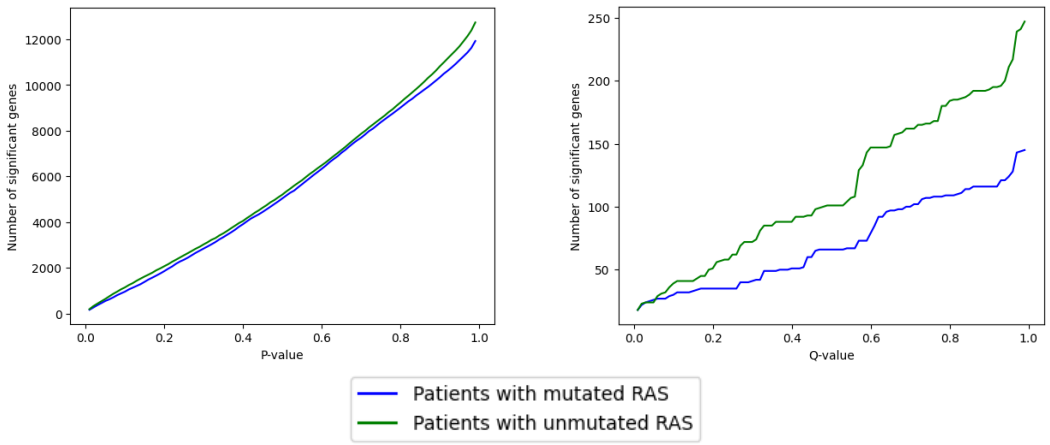


Figure 14. Number of genes chosen by MutSig2CV as the significant threshold varies in the **Training 75% dataset**

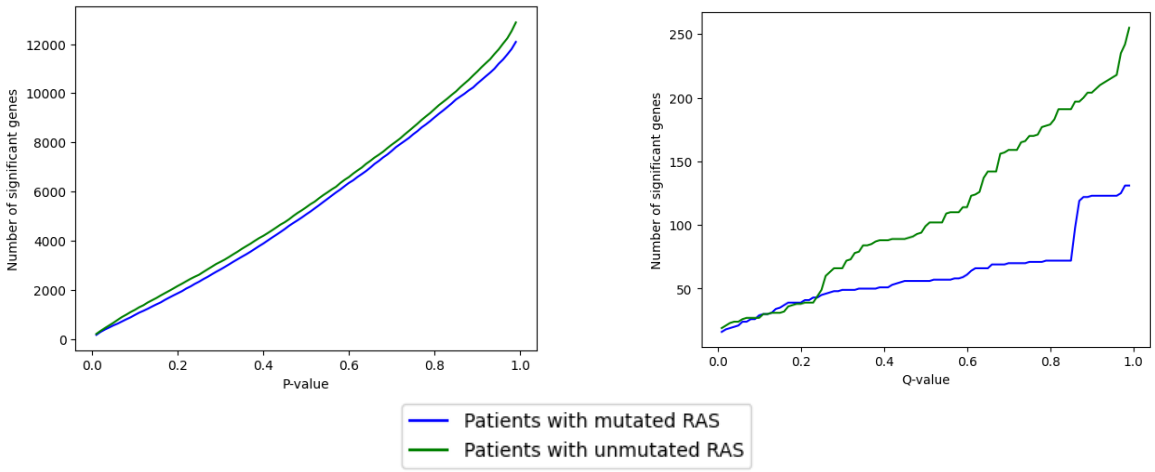


Figure 15. Number of genes chosen by MutSig2CV as the significant threshold varies in the **Training 80% dataset**

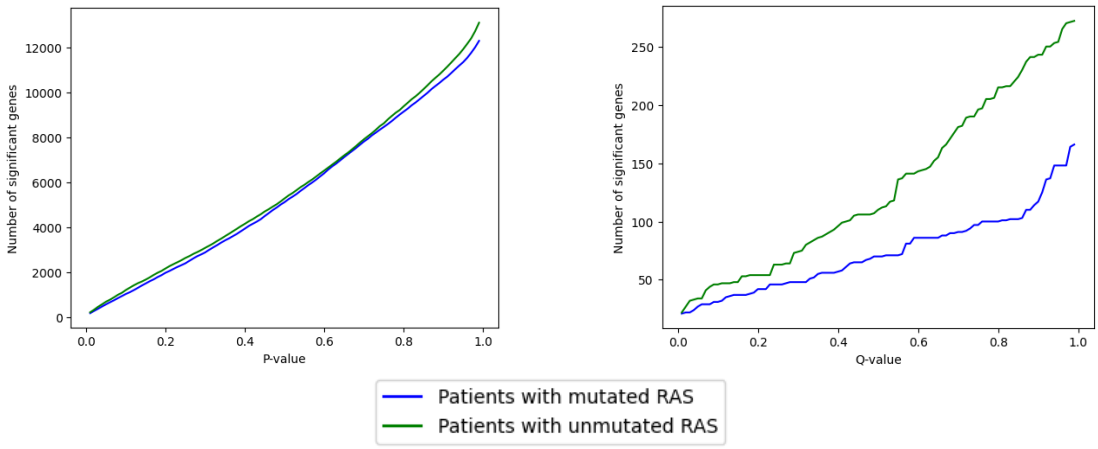


Figure 16. Number of genes chosen by MutSig2CV as the significant threshold varies in the **Training 85% dataset**

According to the documentation, genes with q-value <0.1 indicates that they are significantly mutated. In this study, we decide to continue the analysis for both genes with q-value <0.1 and q-value<1. The choice to keep genes with a q-value < 1, that is not stringent (and thus to accept the possibility of having more false positives), is because we are in the early stage of the study and want to evaluate how the later proposed methods perform with a larger number of mutated genes.

The table shows the number of significantly relevant genes for the four case histories for all three scenarios.

	Training 75 %		**Training 80 %**		**Training 85 %**	
	Q-value < 0,01	**Q-value < 1**	**Q-value < 0,01**	**Q-value < 1**	**Q-value < 0,01**	**Q-value < 1**
Significantly relevant genes exclusive to patients with the mutated RAS gene	116	17	99	16	137	15
Significantly relevant genes exclusive of patients with the unmutated RAS gene.	214	26	226	14	233	30
Significantly relevant genes given by the sum of the two previous exclusive cases.	**330**	**43**	**325**	**30**	**370**	**45**
Significantly relevant genes in common to patients with the mutated and non-mutated RAS gene.	34	13	32	13	44	16

Table 7. Summary of significantly relevant genes for the four cases for all three scenarios

5.3. MutClustSW

The genes selected by MutSig2CV are used to create the input files for MutClust2SW, which is responsible for determining mutational hotspots.

The "*.txt*" files are prepared and given as input to MutClustSW for hotspot selection. In a file, each gene is associated with its length. To obtain this information, the Python library "*pyensembl*" is used, which allows, compared to the previous work that approximated this value, to have the correct value with the possibility of choosing the reference genome (in this case, the hg19/GRCh37 genome that corresponds to version 75). The second "*.txt*" file contains two columns, where the first one is composed of the

gene and the type of mutation (e.g., "APC_Nonsense_Mutation"), the second one contains the position along the protein where there is the mutation.

MutClustSW is run for each group of genes listed above, and a "*.txt*" file is obtained that contains the list of genes with the mutation, the starting and ending position location, and the respective p-value. Also, in this case, significantly relevant hotspots are selected based on the p-value, indicating that the cluster is less likely to be due to chance.

5.3.1. MutClustSW Results

The graphs below show the significant clusters/hotspots as the p-value varies as the four cases (described in 5.2) change.

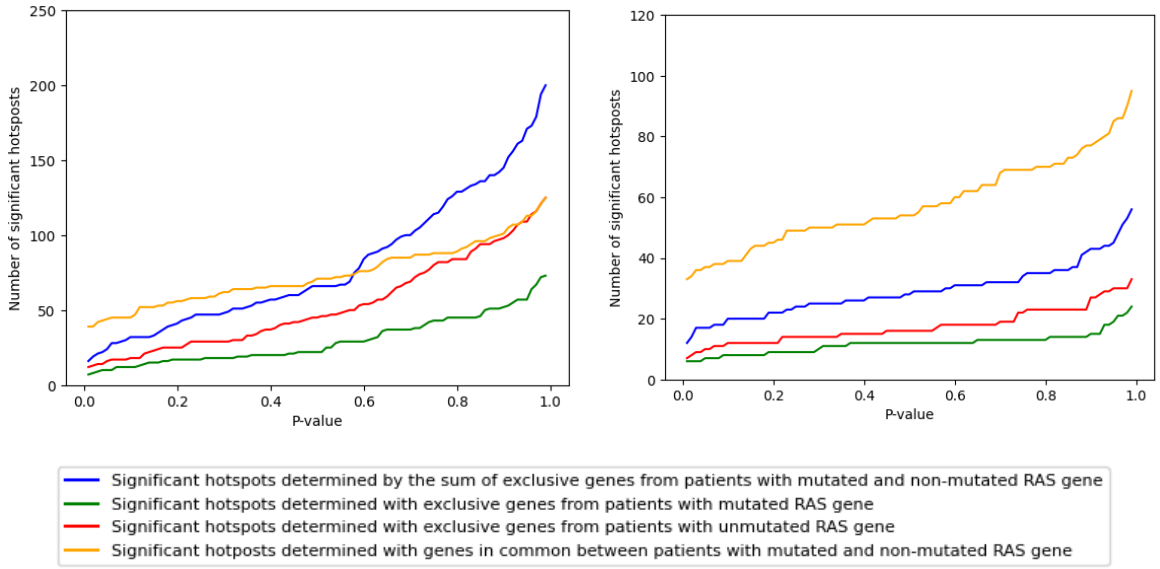


Figure 17. Number of hotspots chosen by MutClustSW as the significant threshold varies in the **Training 75% dataset:** the graph on the left represents the case with q-value <1, while the graph on the right represents the case with q-value<0.01

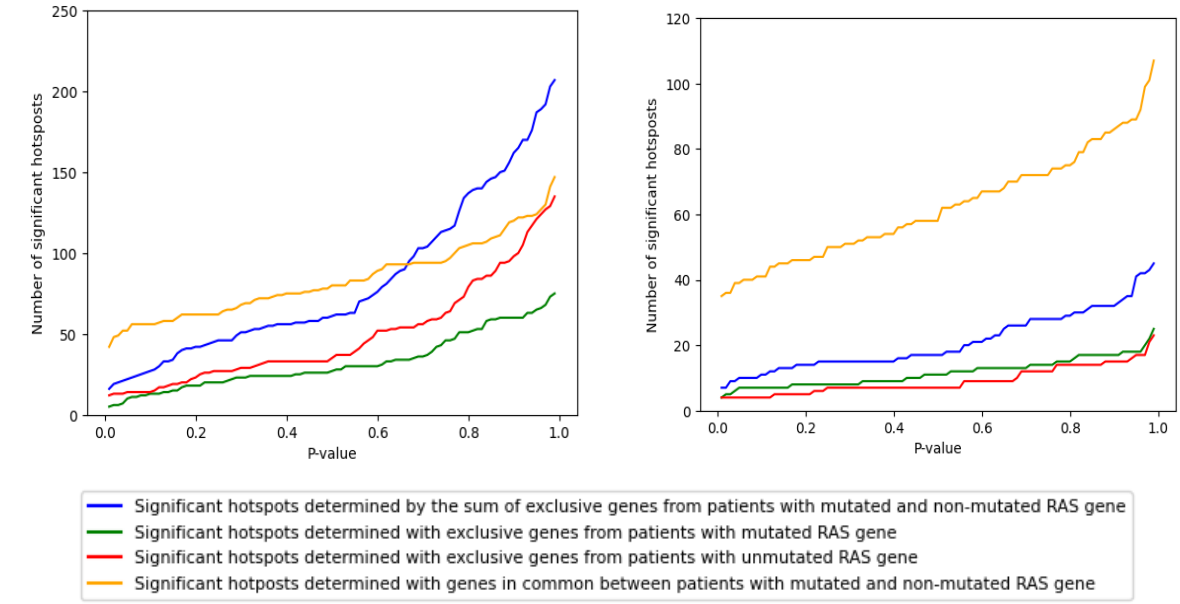


Figure 18. Number of hotspots chosen by MutClustSW as the significant threshold varies in the **Training 80% dataset:** the graph on the left represents the case with q-value <1, while the graph on the right represents the case with q-value<0.01

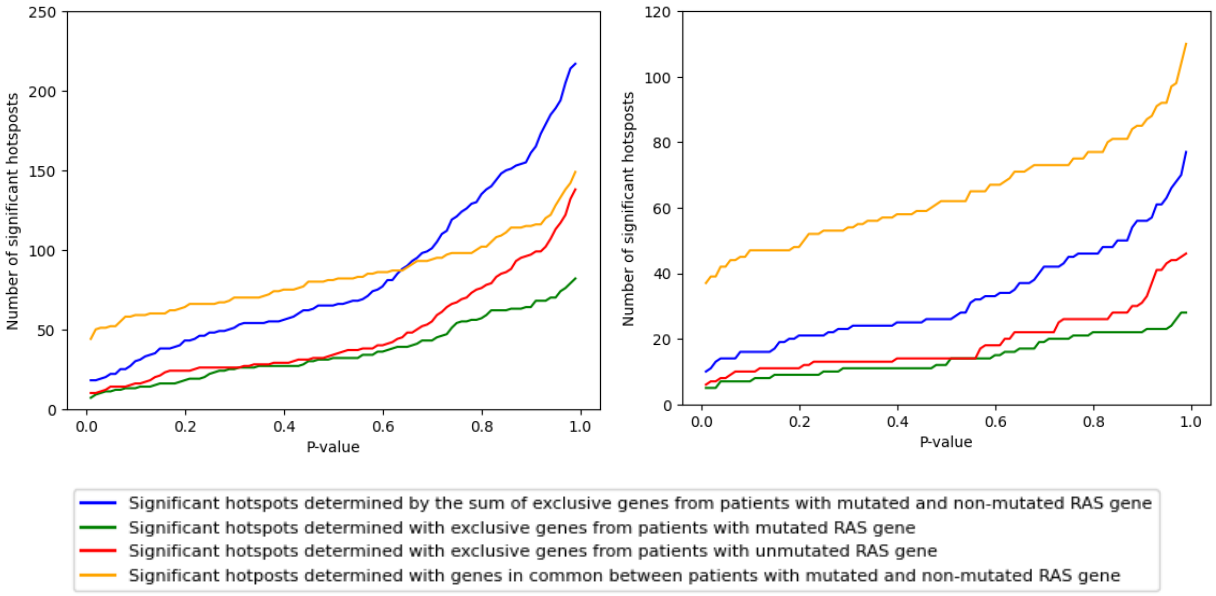


Figure 19. Number of hotspots chosen by MutClustSW as the significant threshold varies in the **Training 85% dataset**: the graph on the left represents the case with q-value <1, while the graph on the right represents the case with q-value<0.01

In MutClustSW documentation, mutational hotspots with p-value<0.05 are defined as significant. In this case, we decide to keep 0.05 as the threshold.

The table below summarise the results for the four gene classes and the three scenarios.

Training 75 %						
Set of genes given as input to MutClustSW	**Number of significant hotspots**		**Exclusive genes**		**Mutation types**	
	Q<0,1	**Q<1**	**Q<0,1**	**Q<1**	**Q<0,1**	**Q<1**
Significantly relevant genes exclusive to patients with the mutated RAS gene	7	10	5	10	2	2
Significantly relevant genes exclusive of patients with the unmutated RAS gene.	10	16	10	16	1	1
Significantly relevant genes given by the sum of the previous two cases.	17	24	15	23	2	2
Significantly relevant genes in common to patients with the mutated and non-mutated RAS gene.	37	44	9	15	4	4

Table 8. Summary of the MutClustSW results for the **Training 75 % dataset**

Training 80 %						
Set of genes given as input to MutClustSW	**Number of significant hotspots**		**Exclusive genes**		**Mutation types**	
	Q<0,1	**Q<1**	**Q<0,1**	**Q<1**	**Q<0,1**	**Q<1**
Significantly relevant genes exclusive to patients with the mutated RAS gene	7	10	5	10	2	2
Significantly relevant genes exclusive of patients with the unmutated RAS gene.	4	14	4	14	1	2
Significantly relevant genes given by the sum of the previous two cases.	10	22	9	20	2	2
Significantly relevant genes in common to patients with the mutated and non-mutated RAS gene.	39	52	11	19	4	4

Table 9. Summary of the MutClustSW results for the **Training 80 % dataset**

Training 85 %						
Set of genes given as input to MutClustSW	**Number of significant hotspots**		**Exclusive genes**		**Mutation types**	
	Q<0,1	**Q<1**	**Q<0,1**	**Q<1**	**Q<0,1**	**Q<1**
Significantly relevant genes exclusive to patients with the mutated RAS gene	7	11	6	11	2	2
Significantly relevant genes exclusive of patients with the unmutated RAS gene.	8	14	10	14	1	1
Significantly relevant genes given by the sum of the previous two cases.	10	22	9	22	2	2
Significantly relevant genes in common to patients with the mutated and non-mutated RAS gene.	42	52	12	18	4	4

Table 10. Summary of the MutClustSW results for the **Training 85 % dataset**

5.4. Matrix Occurrence creation

A matrix of occurrences is created in which the rows are patients and the columns are mutational features/signatures. Specifically, mutations in MutSig-selected genes that belong to a hotspot are labelled as follows: gene, mutation type, initial hotspot position along the protein (by convention, the initial position included), and final hotspot position along the protein (by convention, the final position is excluded), example APC_Nonsense_Mutation_1450_1451. Instead, mutations in MutSig-selected genes that do not belong to any hotspot are labelled as follows: gene, type of mutation, "noclust" (identifies not belonging to any hotspot). In addition, a "target" column is added where at a patient with the mutated RAS gene family "1" is added, otherwise it remains zero.

In this study, the matrix of occurrences is created from the group of significantly relevant genes given by the sum of the exclusive genes of patients with the mutated RAS gene family with the exclusive genes of patients with the unmutated RAS gene

family. The matrix is then filled by counting the mutations in the corresponding columns.

5.4.1. Matrix Occurrence creation: Results

As explained earlier, the columns of the matrix (which are the mutational signatures) are created by selecting the significantly relevant genes given by the sum of the patient-exclusive genes with the RAS mutated and non-mutated gene family. The table below shows the number of mutational signatures in cases with q-value < 0.1 and q-value < 1 in the three scenarios.

	Training 75 %		**Training 80 %**		**Training 85 %**	
	Q<0,1	**Q<1**	**Q<0,1**	**Q<1**	**Q<0,1**	**Q<1**
Number of mutational signatures	184	1145	141	1173	222	1351

Table 11. Number of mutational signature for the three scenarios for the case with q-value <0,1 and q-value <1

The following tables give the details of the mutational signatures: specifically, the number of hotspots, genes, and type of mutations exclusive to both mutational signatures labelled "*clust*" (genes identified by MutClustSW that belong to a hotspot) and "*noclust*" (genes identified by MutSig2CV that do not belong to any hotspot) for the three scenarios.

Training 75 %							
Exclusive genes in hotspot columns		**Exclusive genes in no-hotspot columns**		**Type mutations in hotspot columns**		**Type mutations in no-hotspot columns**	
Q<0,1	**Q<1**	**Q<0,1**	**Q<1**	**Q<0,1**	**Q<1**	**Q<0,1**	**Q<1**
15	23	43	330	2	2	14	17

Training 80 %							
Exclusive genes in hotspot columns		**Exclusive genes in no-hotspot columns**		**Type mutations in hotspot columns**		**Type mutations in no-hotspot columns**	
Q<0,1	**Q<1**	**Q<0,1**	**Q<1**	**Q<0,1**	**Q<1**	**Q<0,1**	**Q<1**
9	20	30	325	2	2	15	17

Training 85 %							
Exclusive genes in hotspot columns		**Exclusive genes in no-hotspot columns**		**Type mutations in hotspot columns**		**Type mutations in no-hotspot columns**	
Q<0,1	**Q<1**	**Q<0,1**	**Q<1**	**Q<0,1**	**Q<1**	**Q<0,1**	**Q<1**
9	22	45	370	2	2	16	17

Table 12. Details of the mutations signatures for the three scenarios

Finally, this table summarizes the sizes of the occurrence matrices for the three scenarios:

	Training 75 %		**Training 80 %**		**Training 85 %**	
	Q-value < 0,1	**Q-value < 1**	**Q-value < 0,1**	**Q-value < 1**	**Q-value < 0,1**	**Q-value < 1**
Rows (Patients)	755	755	805	805	854	854
Columns (Mutational Signatures)	184	1145	141	1173	222	1351

Table 13. Summary of the details of the occurrence matrices for the three scenarios

5.5. Statistical and Machine Learning models

Supervised Machine Learning approaches are used to determine which mutational features/signatures are characteristic of the patients of the group under study (in this case, patients with mutated RAS gene family).

To this aim, a further preprocessing of the matrices is made: since they are not balanced in terms of patients with the mutated RAS gene family (40%) and patients with the unmutated RAS gene family (60%), to better evaluate the performance of the models, patients with no mutations in the RAS gene family are undersampled.

Specifically, we obtain three training and three testing sets, each one including 50% of patients with the mutated RAS gene family and 50 % of patients with unmutated RAS gene family. These sets are originated from previous scenarios (Training 75%, Training 80%, Training 85%) simply randomly selecting for each case the same amount of RAS-unmutated patients as the number of RAS-mutated ones. In this way, the percentages of training/testing splits are different from those of the three previous scenarios, but the selected training and testing patients are perfect subsets of the previous ones to provide better comparability.

The following tables show the number of patients with the mutated RAS gene family labeled with 1 and the unmutated RAS gene family labeled with 0 in the balanced and unbalanced versions of the datasets for the three different scenarios.

From Training 75 %							
Balanced				**Unbalanced**			
Training		**Test**		**Training 75 %**		**Test 25 %**	
0	**1**	**0**	**1**	**0**	**1**	**0**	**1**
302	302	76	75	453	302	113	76

Training 80 %							
Balanced				**Unbalanced**			
Training 80 %		**Test 20 %**		**Training 80 %**		**Test 20 %**	
0	**1**	**0**	**1**	**0**	**1**	**0**	**1**
322	322	65	64	483	322	97	64

Training 85 %							
Balanced				**Unbalanced**			
Training 85 %		**Test 15 %**		**Training 85 %**		**Test 15 %**	
0	**1**	**0**	**1**	**0**	**1**	**0**	**1**
342	342	52	51	512	342	77	51

Table 14. Number of patients labeled with 0 and 1 in case they belong to the category of patients with the RAS mutated or not mutated gene family respectively in the balanced and unbalanced datasets for the three scenarios.

The models are then trained on the training sets, while the performance of the models is evaluated using the test set.

Lasso Logistic regression is used to decrease the number of features while training a classifier to recognize RAS-mutated cases and estimating the performance in terms of accuracy, precision, recall and f1-score. As described in 4.3.1, Lasso Logistic Regression forces some model coefficients to 0, working as a selection of mutational features/signatures. The results obtained from this model, are then compared with two other methodologies fed with the same input features, as shown below.

In the first methodology, a Bootstrapping approach of 100 Lasso Logistic Regression models is performed for two different purposes. First, to evaluate the robustness of the first individual Lasso Logistic Regression by producing rankings based on the number of times features are selected by Bootstrapping and see which position is taken by

those selected by Lasso Logistic Regression. Second, to perform further feature selection preserving only features associated with a null coefficient less than a certain number of times among the 100 models. So-selected features are then simultaneously used for other three models, a full (not-regularized) Logistic Regression, a Ridge-regularized Logistic Regression and a Random Forest. Moreover, each model is evaluated in terms of accuracy, precision, recall and f1-score.

A second methodology is proposed as an alternative selection option based on feature importance: in particular, feature selection is made after applying Mean Decrease in Impurity. In this way, using different techniques and comparing their results, could be useful to better determine features that are preserved in the methods proposed and to determine the robustness of the model used. As in the previous case, the selected features are simultaneously pulled into the Logistic Regression, Ridge Logistic Regression and Random Forest models and evaluated in terms of accuracy, precision, recall and f1-score.

5.5.1. Lasso Logistic Regression model Results

The first model proposed in this study is the Lasso Logistic Regression. The following are its performance values in terms of accuracy, precision, recall, F1-score and the number of features selected (i.e., the feature with coefficients different from zero) by the model itself.

Lasso Logistic Regression - Training 75 %									
Balanced				**Q-value < 0,1**	**Unbalanced**				**Q-value < 0,1**
	Prec.	**Recall**	**F1-s.**	**Num. Feat**		**Prec.**	**Recall**	**F1-s.**	**Num. Feat**
0	0,57	0,86	0,68	118	**0**	0,69	0,90	0,78	144
1	0,70	0,35	0,46		**1**	0,73	0,39	0,51	
Acc.	0,60				**Acc.**	0,70			

Balanced				Q-value < 1	Unbalanced				Q-value < 1
	Prec.	Recall	F1-s.	Num. Feat		Prec.	Recall	F1-s	Num. Feat
0	0,55	0,63	0,59	334	**0**	0,66	0,78	0,72	403
1	0,56	0,49	0,53		**1**	0,55	0,41	0,47	
Acc.	0,56				**Acc.**	0,63			

Lasso Logistic Regression - Training 85 %									
Balanced				**Q-value < 0,1**	**Unbalanced**				**Q-value < 0,1**
	Prec.	**Recall**	**F1-s.**	**Num. Feat**		**Prec.**	**Recall**	**F1-s.**	**Num. Feat**
0	0,56	0,77	0,65	158	**0**	0,62	0,82	0,70	174
1	0,62	0,39	0,48		**1**	0,48	0,25	0,33	
Acc.	0,58				**Acc.**	0,59			
Balanced				**Q-value < 1**	**Unbalanced**				**Q-value < 1**
	Prec.	**Recall**	**F1-s.**	**Num. Feat**		**Prec.**	**Recall**	**F1-s**	**Num. Feat**
0	0,56	0,65	0,60	402	**0**	0,64	0,79	0,71	480
1	0,57	0,47	0,52		**1**	0,53	0,35	0,42	
Acc.	0,56				**Acc.**	0,61			

Lasso Logistic Regression - Training 80 %									
Balanced				**Q-value < 0,1**	**Unbalanced**				**Q-value < 0,1**
	Prec.	**Recall**	**F1-s.**	**Num. Feat**		**Prec.**	**Recall**	**F1-s.**	**Num. Feat**
0	0,58	0,94	0,72	97	**0**	0,60	0,88	0,71	107
1	0,83	0,31	0,45		**1**	0,40	0,12	0,19	
Acc.	0,63				**Acc.**	0,58			

Balanced				Q-value < 1	Unbalanced				Q-value < 1
	Prec.	Recall	F1-s.	Num. Feat		Prec.	Recall	F1-s	Num. Feat
0	0,59	0,63	0,61	364	**0**	0,66	0,77	0,71	422
1	0,60	0,56	0,58		**1**	0,53	0,39	0,45	
Acc.	0,60				**Acc.**	0,62			

Table 15. Results of the Lasso Logistic Regression Model for the three scenarios.

The performance results obtained from this model are compared with the results obtained from the following two proposed pipelines.

5.5.1.1. Results evaluation

After a first analysis of the three scenarios, attention is turned to the case with the selection of genes from MutSig2CV with q-value < 0.1 and balanced, as they show overall better performance on the class of interest 1. In particular, the choice of performing the analysis by considering the sum of exclusive genes belonging to the category of patients with the mutated and non-mutated RAS gene family allows us to evaluate the selected features by the Lasso Logistic Regression method in the following way: the number of genes belonging to the category of patients with the mutated RAS gene family, which features belong to these genes, and finally the number of patients having these features.

Next, the results for the three scenarios are reported.

	Training 75 % q-value < 0,1 Balanced	**Training 80 % q-value < 0,1 Balanced**	**Training 85 % q-value < 0,1 Balanced**
Features selected by Lasso Logistic Regression	118	97	158
Total genes belonging to the category of patients with the mutated RAS gene family	43	30	45

Total patients belonging to the category of patients with the mutated RAS gene family	302	322	342
Genes belonging to the category of patients with the mutated RAS gene family in the features selected by Lasso Logistic Regression	17 (40%)	16 (53 %)	15 (33 %)
Features that contain the genes that belong to the category of patients with the mutated RAS gene family in the features selected by Lasso Logistic Regression	49 (42 %)	55 (57 %)	62 (39 %)
Patients who have the features that contain the genes belonging to the category of patients with the mutated RAS gene family in the features selected by Lasso Logistic Regression	182 (60 %)	162 (50 %)	232 (68 %)

Table 16. Details of genes, features and patients selected by Lasso Logistic Regression

In addition, we can also evaluate which features and genes belonging to the category of patients with the mutated RAS gene family, shown in the table above, are in common in the three scenarios.

	In common between the Training 75 % and Training 80 %	**In common between the Training 75 % and Training 85 %**	**In common between the Training 80 % and Training 85 %**	**In common between the three scenarios**
Features	22	22	26	15
Genes	8	7	8	5

Table 17. Features and genes analysis

5.5.2. First method proposal Results

As anticipated in section 4.3.3, one of the roles of the Bootstrapping method is feature selection. In particular, Bootstrapping method is performed using 100 Lasso Logistic Regression models trained on random sampling of the 75, 80 and 85 percent of the training data (respectively for the scenarios Training 75 %, Training 80 % and Training 85%).

In order to identify the most relevant features, we perform feature bootstrapping and analyse the frequency distribution of nonzero coefficients obtained from 100 iterations.

To give an example, we plot the distribution of number of features based on the number of times out of 100 that each of them has non-zero coefficient of the Training 75 % scenario in the unbalanced case and gene selection with q-value < 1. The x-axis represents the number of times a feature has nonzero coefficients across the 100 bootstrapping iterations, while the y-axis represents the frequency with which the number of times feature have non-zero coefficients occur across the 100-bootstrapping iteration.

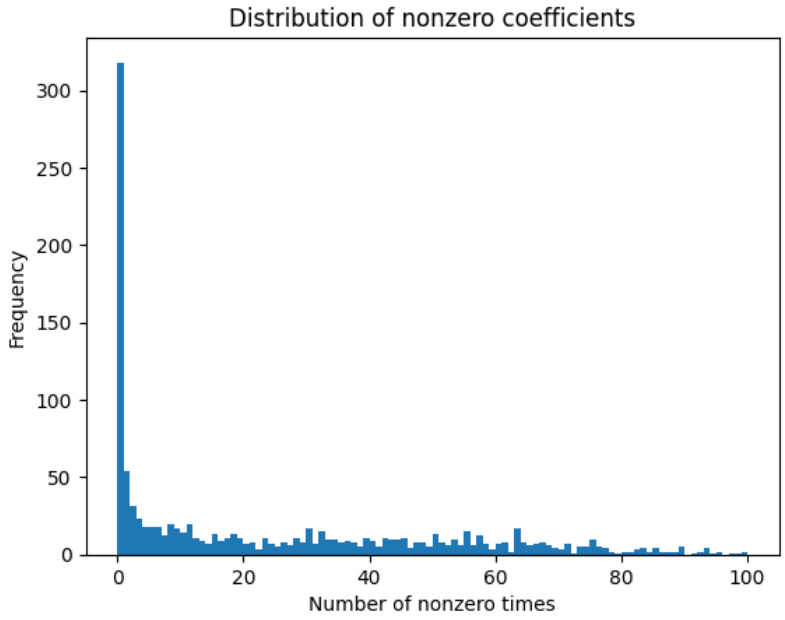


Figure 20. Distribution of number of times features have non-zero coefficients in the bootstrapping iterations and the number of features that occur with that number of times.

So, if a feature has a non-zero coefficient in 20 out of the 100 bootstrapping iterations, the corresponding value on the x-axis will be 20. If there are 10 features that have non-zero coefficients in 20 out of the 100 bootstrapping iterations, the corresponding value on the y-axis will be 10.

In essence, the plots show the distribution of number of times features have non-zero coefficients in the bootstrapping iterations and the number of features that occur with that number of times.

Since the graphs do not suggest a threshold for selecting features, we analyse the frequency distribution of nonzero coefficients obtained from feature bootstrapping in terms of percentiles. The percentile indicates the percentage of times a feature has a non-zero coefficient in the 100 iterations of the bootstrap. The percentiles, the number of times the features have a non-zero coefficient, and the number of features is given in the following table.

Bootstrapping with Lasso Logistic Regression – Training 75 %														
Balanced				**Q-value < 0,1**				**Unbalanced**				**Q-value < 0,1**		
Perc.	30	40	50	60	70	80	90	**Perc.**	30	40	50	60	70	80	90
N. times	17	54	62	69	84	90	96	**N. times**	56	63	67	76	87	92	97
N. feat.	129	111	97	74	56	39	20	**N. feat.**	130	112	96	75	58	38	19
Balanced				**Q-value <1**				**Unbalanced**				**Q-value < 1**		
Perc.	30	40	50	60	70	80	90	**Perc.**	30	40	50	60	70	80	90
N. times	0	3	10	19	29	47	60	**N. times**	1	5	12	25	37	51	66
N. feat.	1146	692	574	466	351	231	117	**N. feat.**	818	967	581	467	351	239	116

Bootstrapping with Lasso Logistic Regression – Training 80 %														
Balanced				**Q-value < 0,1**				**Unbalanced**				**Q-value < 0,1**		
Perc.	30	40	50	60	70	80	90	**Perc.**	30	40	50	60	70	80	90
N. times	58	63	67	82	89	93	96	**N. times**	52	62	65	70	78	91	95
N. feat.	101	87	72	57	44	34	15	**N. feat.**	99	87	75	61	46	29	16
Balanced				**Q-value <1**				**Unbalanced**				**Q-value < 1**		
Perc.	30	40	50	60	70	80	90	**Perc.**	30	40	50	60	70	80	90
N. times	0	4	10	21	34	47	63	**N. times**	2	7	15	25	39	53	66
N. feat.	1174	706	600	472	353	235	121	**N. feat.**	842	717	594	481	353	242	121

Bootstrapping with Lasso Logistic Regression – Training 85 %														
Balanced				**Q-value < 0,1**				**Unbalanced**				**Q-value < 0,1**		
Perc.	30	40	50	60	70	80	90	**Perc.**	30	40	50	60	70	80	90
N. times	43	55	64	71	86	92	96	**N. times**	58	64	68	80	89	94	97
N. feat.	156	134	114	93	67	48	28	**N. feat.**	158	134	112	89	71	49	24
Balanced				**Q-value <1**				**Unbalanced**				**Q-value < 1**		
Perc.	30	40	50	60	70	80	90	**Perc.**	30	40	50	60	70	80	90
N. times	0	3	8	16	30	46	64	**N. times**	1	5	13	26	37	52	66
N. feat.	1352	819	690	555	413	271	138	**N. feat.**	1002	822	692	545	412	276	136

Table 18. Results of the Bootstrapping with Lasso Logistic Regression for the three scenarios

By looking at the results obtained, we decide to select the number of features not according to a fixed percentile because it would lead to unbalanced and reduced numerosities among the scenarios and subcases that would make the comparison between them inconsistent. We, therefore, decide to select features based on the number of times the features have a non-zero coefficient: in this thesis, we chose features with the non-zero coefficient at least 50 % of the times out of the 100 Lasso Logistic Models performed by Bootstrapping.

Bootstrapping – Lasso Logistic Regression								
Training 75 %								
	Balanced	**Q<0,1**	**Unbalanced**	**Q<0,1**	**Balanced**	**Q<1**	**Unbalanced**	**Q<1**
Perc.	40		30		90		80	
N. times	54		56		60		51	
N. feat.	111		130		117		239	
Training 80 %								
	Balanced	**Q<0,1**	**Unbalanced**	**Q<0,1**	**Balanced**	**Q<1**	**Unbalanced**	**Q<1**
Perc.	30		30		90		80	
N. times	58		52		63		53	
N. feat.	101		99		121		242	
Training 85 %								
	Balanced	**Q<0,1**	**Unbalanced**	**Q<0,1**	**Balanced**	**Q<1**	**Unbalanced**	**Q<1**
Perc.	40		30		90		80	
N. times	55		58		64		52	
N. feat.	134		158		138		276	

Table 19. Summary of the Bootstrapping with Lasso Logistic Regression for the three scenarios

After that, the performance of three models with the newly selected features is evaluated: Logistic Regression, Ridge Logistic Regression and Random Forest.

Below, the results are reported.

First Method - Logistic Regression – Training 75 %							
Balanced		**Q-value < 0,1**		**Unbalanced**		**Q-value < 0,1**	
	Precision	**Recall**	**F1-score**		**Precision**	**Recall**	**F1-score**
0	0,54	0,84	0,66	**0**	0,66	0,90	0,76
1	0,62	0,27	0,37	**1**	0,69	0,32	0,43
Accuracy	0,56			**Accuracy**	0,67		
Balanced		**Q-value < 1**		**Unbalanced**		**Q-value < 1**	
	Precision	**Recall**	**F1-Score**		**Precision**	**Recall**	**F1-Score**
0	0,53	0,82	0,65	**0**	0,62	0,80	0,70
1	0,60	0,28	0,43	**1**	0,49	0,29	0,36
Accuracy	0,55			**Accuracy**	0,59		

First Method - Logistic Regression – Training 80 %							
Balanced		**Q-value < 0,1**		**Unbalanced**		**Q-value < 0,1**	
	Precision	**Recall**	**F1-score**		**Precision**	**Recall**	**F1-score**
0	0,57	0,86	0,70	**0**	0,59	0,86	0,70
1	0,75	0,33	0,46	**1**	0,33	0,11	0,16
Accuracy	0,61			**Accuracy**	0,56		
Balanced		**Q-value < 1**		**Unbalanced**		**Q-value < 1**	
	Precision	**Recall**	**F1-Score**		**Precision**	**Recall**	**F1-Score**
0	0,61	0,79	0,69	**0**	0,52	0,74	0,61
1	0,43	0,23	0,30	**1**	0,53	0,30	0,38
Accuracy	0,57			**Accuracy**	0,52		

First Method - Logistic Regression – Training 85 %							
Balanced		**Q-value < 0,1**		**Unbalanced**		**Q-value < 0,1**	
	Precision	**Recall**	**F1-score**		**Precision**	**Recall**	**F1-score**
0	0,53	0,88	0,67	**0**	0,62	0,88	0,73
1	0,65	0,22	0,32	**1**	0,53	0,19	0,28
Accuracy	0,55			**Accuracy**	0,60		
Balanced		**Q-value < 1**		**Unbalanced**		**Q-value < 1**	
	Precision	**Recall**	**F1-Score**		**Precision**	**Recall**	**F1-Score**
0	0,50	0,79	0,61	**0**	0,60	0,86	0,71
1	0,48	0,20	0,28	**1**	0,42	0,15	0,23
Accuracy	0,50			**Accuracy**	0,57		

Table 20. Results of the Logistic Regression for the three scenarios in the first method proposed

First Method - Ridge Logistic Regression – Training 75 %							
Balanced		**Q-value < 0,1**		**Unbalanced**		**Q-value < 0,1**	
	Precision	**Recall**	**F1-score**		**Precision**	**Recall**	**F1-score**
0	0,53	0,84	0,65	**0**	0,65	0,92	0,76
1	0,61	0,25	0,36	**1**	0,69	0,26	0,38
Accuracy	0,55			**Accuracy**	0,66		
Balanced		**Q-value < 1**		**Unbalanced**		**Q-value < 1**	
	Precision	Recall	F1-Score		Precision	Recall	F1-Score
0	0,53	0,82	0,65	**0**	0,62	0,78	0,69
1	0,60	0,28	0,38	**1**	0,47	0,29	0,36
Accuracy	0,55			**Accuracy**	0,58		

First Method - Ridge Logistic Regression – Training 80 %							
Balanced		**Q-value < 0,1**		**Unbalanced**		**Q-value < 0,1**	
	Precision	**Recall**	**F1-score**		**Precision**	**Recall**	**F1-score**
0	0.59	0,92	0,72	**0**	0,59	0,86	0,70
1	0,81	0,34	0,48	**1**	0,33	0,11	0,16
Accuracy	0,64			**Accuracy**	0,56		
Balanced		**Q-value < 1**		**Unbalanced**		**Q-value < 1**	
	Precision	Recall	F1-Score		Precision	Recall	F1-Score
0	0,51	0,72	0,60	**0**	0,61	0,80	0,69
1	0,51	0,30	0,38	**1**	0,42	0,22	0,29
Accuracy	0,51			**Accuracy**	0,57		

First Method - Ridge Logistic Regression – Training 85 %							
Balanced		**Q-value < 0,1**		**Unbalanced**		**Q-value < 0,1**	
	Precision	**Recall**	**F1-score**		**Precision**	**Recall**	**F1-score**
0	0,53	0,87	0,66	**0**	0,62	0,88	0,73
1	0,61	0,22	0,32	**1**	0,53	0,19	0,28
Accuracy	0,54			**Accuracy**	0,60		
Balanced		**Q-value < 1**		**Unbalanced**		**Q-value < 1**	
	Precision	Recall	F1-Score		Precision	Recall	F1-Score
0	0,51	0,81	0,62	**0**	0,60	0,87	0,71
1	0,50	0,20	0,28	**1**	0,44	0,15	0,23
Accuracy	0,50			**Accuracy**	0,58		

Table 21. Results of the Ridge Logistic Regression for the three scenarios in the first method

First Method - Random Forest – Training 75 %							
Balanced		**Q-value < 0,1**		**Unbalanced**		**Q-value < 0,1**	
	Precision	**Recall**	**F1-score**		**Precision**	**Recall**	**F1-score**
0	0,54	0,80	0,64	**0**	0,65	0,88	0,75
1	0,59	0,29	0,39	**1**	0,62	0,28	0,38
Accuracy	0,64			**Accuracy**	0,53		
Balanced		**Q-value < 1**		**Unbalanced**		**Q-value < 1**	
	Precision	Recall	F1-Score		Precision	Recall	F1-Score
0	0,52	0,80	0,63	**0**	0,65	0,83	0,73
1	0,56	0,25	0,35	**1**	0,57	0,33	0,42
Accuracy	0,53			**Accuracy**	0,63		

First Method - Random Forest – Training 80 %							
Balanced		**Q-value < 0,1**		**Unbalanced**		**Q-value < 0,1**	
	Precision	**Recall**	**F1-score**		**Precision**	**Recall**	**F1-score**
0	0,56	0,88	0,69	**0**	0,60	0,88	0,71
1	0,71	0,31	0,43	**1**	0,40	0,12	0,19
Accuracy	0,60			**Accuracy**	0,58		
Balanced		**Q-value < 1**		**Unbalanced**		**Q-value < 1**	
	Precision	Recall	F1-Score		Precision	Recall	F1-Score
0	0,51	0,78	0,62	**0**	0,61	0,84	0,70
1	0,52	0,23	0,32	**1**	0,43	0,19	0,26
Accuracy	0,51			**Accuracy**	0,58		

First Method - Random Forest – Training 85 %							
Balanced		**Q-value < 0,1**		**Unbalanced**		**Q-value < 0,1**	
	Precision	**Recall**	**F1-score**		**Precision**	**Recall**	**F1-score**
0	0,52	0,79	0,63	**0**	0,62	0,81	0,70
1	0,54	0,25	0,35	**1**	0,48	0,27	0,35
Accuracy	0,52			**Accuracy**	0,59		
Balanced		**Q-value < 1**		**Unbalanced**		**Q-value < 1**	
	Precision	Recall	F1-Score		Precision	Recall	F1-Score
0	0,52	0,79	0,63	**0**	0,62	0,86	0,72
1	0,54	0,25	0,35	**1**	0,52	0,23	0,32
Accuracy	0,52			**Accuracy**	0,60		

Table 22. Results of the Random Forest for the three scenarios in the first method

5.5.2.1. Result evaluation and mutational feature prioritization

The second role covered by the Bootstrapping method is the evaluation of the robustness of individual models in the Lasso Logistic Regression. Specifically, the features are ranked in descending order according to the number of times they are selected in the Bootstrapping model. In terms of robustness, we consider the features in common in the three downstream scenarios (in the case with q-value < 0,1 and balanced) in the first proposed method, i.e., Lasso Logistic regression, to determine their position within the ranking made by the Bootstrapping model. In total, 15 features are in common and below are reported their ranking obtained in the three scenarios.

Training 75 % / q-value < 0,1 / Balanced	
Feature	**Ranking**
TGIF1_Nonsense_Mutation_noClust	11
PTEN_Splice_Site_noClust	16
ERBB2_Splice_Region_noClust	32
TGIF1_Splice_Site_noClust	52
TGIF1_Frame_Shift_Ins_noClust	58
ERBB2_Missense_Mutation_noClust	63
ELF3_Frame_Shift_Ins_noClust	87
ERBB2_Silent_noClust	94
PTEN_Nonsense_Mutation_noClust	98
ELF3_Missense_Mutation_noClust	113
ELF3_Nonsense_Mutation_noClust	120
TGIF1_Missense_Mutation_noClust	125
BCL9_Splice_Site_noClust	165
BCL9_Nonsense_Mutation_noClust	166
PTEN_Missense_Mutation_noClust	175

Training 80 % / q-value < 0,1 / Balanced	
Feature	**Ranking**
TGIF1_Splice_Site_noClust	12
ERBB2_Splice_Region_noClust	14
TGIF1_Missense_Mutation_noClust	20
BCL9_Nonsense_Mutation_noClust	26
ELF3_Frame_Shift_Ins_noClust	35

PTEN_Splice_Site_noClust	38
PTEN_Nonsense_Mutation_noClust	43
TGIF1_Nonsense_Mutation_noClust	45
ERBB2_Missense_Mutation_noClust	78
PTEN_Missense_Mutation_noClust	80
TGIF1_Frame_Shift_Ins_noClust	82
ERBB2_Silent_noClust	93
ELF3_Missense_Mutation_noClust	99
ELF3_Nonsense_Mutation_noClust	109
BCL9_Splice_Site_noClust	125

Training 85 % / q-value < 0,1 / Balanced	
Feature	**Ranking**
PTEN_Splice_Site_noClust	3
TGIF1_Missense_Mutation_noClust	14
ELF3_Missense_Mutation_noClust	15
PTEN_Nonsense_Mutation_noClust	25
ERBB2_Splice_Region_noClust	34
TGIF1_Nonsense_Mutation_noClust	50
TGIF1_Splice_Site_noClust	62
ERBB2_Silent_noClust	92
PTEN_Missense_Mutation_noClust	103
ELF3_Frame_Shift_Ins_noClust	105
ELF3_Nonsense_Mutation_noClust	120
ERBB2_Missense_Mutation_noClust	142

BCL9_Splice_Site_noClust	143
TGIF1_Frame_Shift_Ins_noClust	153
BCL9_Splice_Site_noClust	190

Table 23. Feature in common and their ranking

5.5.3. Alternative feature selection and classification method

The second methodology, proposed and illustrated before, consists of feature selection from the Random Forest method with Feature Importance, called also Mean Decrease in Impurity. As in the case above, the selected features are used to train Logistic Regression, Ridge Logistic Regression and Random Forest models.

After using the Random Forest with Feature Importance, the mutational features are put in ascending order according to their relative importance (i.e., the reduction of the mean is evaluated when a particular feature is used to separate the two groups during the construction of the decision tree: the greater the reduction, the greater the importance of the feature). Once put in order from most important to least important, the numbers selected with the previous Bootstrapping methodology are considered to select the features and better compare the two options.

In the following, the performance results in terms of accuracy, precision, recall and f1-score of the three models (Logistic Regression, Ridge Logistic Regression and Random Forest) are reported.

Second Method - Logistic Regression – Training 75 %							
Balanced		**Q-value < 0,1**		**Unbalanced**		**Q-value < 0,1**	
	Precision	**Recall**	**F1-score**		**Precision**	**Recall**	**F1-score**
0	0,57	0,83	0,68	**0**	0,68	0,89	0,77
1	0,68	0,37	0,48	**1**	0,71	0,38	0,50
Accuracy	0,60			**Accuracy**	0,69		
Balanced		**Q-value < 1**		**Unbalanced**		**Q-value < 1**	
	Precision	**Recall**	**F1-Score**		**Precision**	**Recall**	**F1-Score**
0	0,60	0,64	0,62	**0**	0,65	0,73	0,69
1	0,61	0,56	0,58	**1**	0,52	0,42	0,46
Accuracy	0,60			**Accuracy**	0,61		

Second Method - Logistic Regression – Training 80 %							
Balanced		**Q-value < 0,1**		**Unbalanced**		**Q-value < 0,1**	
	Precision	**Recall**	**F1-score**		**Precision**	**Recall**	**F1-score**
0	0,57	0,91	0,70	**0**	0,59	0,84	0,69
1	0,76	0,30	0,43	**1**	0,33	0,12	0,18
Accuracy	0,60			**Accuracy**	0,55		
Balanced		**Q-value < 1**		**Unbalanced**		**Q-value < 1**	
	Precision	**Recall**	**F1-Score**		**Precision**	**Recall**	**F1-Score**
0	0,48	0,57	0,52	**0**	0,64	0,78	0,70
1	0,46	0,38	0,41	**1**	0,51	0,35	0,41
Accuracy	0,47			**Accuracy**	0,60		

Second Method - Logistic Regression – Training 85 %							
Balanced		**Q-value < 0,1**		**Unbalanced**		**Q-value < 0,1**	
	Precision	**Recall**	**F1-score**		**Precision**	**Recall**	**F1-score**
0	0,56	0,77	0,65	**0**	0,55	0,73	0,63
1	0,62	0,39	0,48	**1**	0,59	0,39	0,47
Accuracy	0,58			**Accuracy**	0,56		
Balanced		**Q-value < 1**		**Unbalanced**		**Q-value < 1**	
	Precision	**Recall**	**F1-Score**		**Precision**	**Recall**	**F1-Score**
0	0,56	0,77	0,65	**0**	0,64	0,79	0,71
1	0,61	0,37	0,46	**1**	0,53	0,35	0,41
Accuracy	0,57			**Accuracy**	0,61		

Table 24. Results of the Logistic Regression for the three scenarios in the second method

Second Method - Ridge Logistic Regression – Training 75 %							
Balanced		**Q-value < 0,1**		**Unbalanced**		**Q-value < 0,1**	
	Precision	**Recall**	**F1-score**		**Precision**	**Recall**	**F1-score**
0	0,57	0,87	0,69	**0**	0,70	0,93	0,80
1	0,71	0,33	0,45	**1**	0,79	0,39	0,53
Accuracy	0,60			**Accuracy**	0,71		
Balanced		**Q-value < 1**		**Unbalanced**		**Q-value < 1**	
	Precision	Recall	F1-Score		Precision	Recall	F1-Score
0	0,58	0,67	0,62	**0**	0,65	0,75	0,7
1	0,60	0,51	0,55	**1**	0,52	0,39	0,45
Accuracy	0,59			**Accuracy**	0,61		

Second Method - Ridge Logistic Regression – Training 80 %							
Balanced		**Q-value < 0,1**		**Unbalanced**		**Q-value < 0,1**	
	Precision	**Recall**	**F1-score**		**Precision**	**Recall**	**F1-score**
0	0,58	0,94	0,72	**0**	0,60	0,84	0,70
1	0,83	0,31	0,45	**1**	0,38	0,16	0,22
Accuracy	0,63			**Accuracy**	0,57		
Balanced		**Q-value < 1**		**Unbalanced**		**Q-value < 1**	
	Precision	**Recall**	**F1-Score**		**Precision**	**Recall**	**F1-Score**
0	0,48	0,58	0,53	**0**	0,69	0,77	0,73
1	0,46	0,36	0,40	**1**	0,58	0,48	0,53
Accuracy	0,47			**Accuracy**	0,66		

Second Method - Ridge Logistic Regression – Training 85 %							
Balanced		**Q-value < 0,1**		**Unbalanced**		**Q-value < 0,1**	
	Precision	**Recall**	**F1-score**		**Precision**	**Recall**	**F1-score**
0	0,56	0,77	0,65	**0**	0,60	0,81	0,69
1	0,61	0,37	0,46	**1**	0,61	0,21	0,28
Accuracy	0,57			**Accuracy**	0,57		
Balanced		**Q-value < 1**		**Unbalanced**		**Q-value < 1**	
	Precision	Recall	F1-Score		Precision	Recall	F1-Score
0	0,56	0,75	0,64	**0**	0,64	0,81	0,71
1	0,61	0,39	0,48	**1**	0,53	0,33	0,4
Accuracy	0,57			**Accuracy**	0,61		

Table 25. Results of the Ridge Logistic Regression for the three scenarios in the second method

Second Method - Random Forest – Training 75 %							
Balanced		**Q-value < 0,1**		**Unbalanced**		**Q-value < 0,1**	
	Precision	**Recall**	**F1-score**		**Precision**	**Recall**	**F1-score**
0	0,57	0,76	0,65	**0**	0,68	0,88	0,77
1	0,63	0,41	0,50	**1**	0,68	0,39	0,50
Accuracy	0,59			**Accuracy**	0,68		
Balanced		**Q-value < 1**		**Unbalanced**		**Q-value < 1**	
	Precision	**Recall**	**F1-Score**		**Precision**	**Recall**	**F1-Score**
0	0,60	0,66	0,63	**0**	0,68	0,85	0,75
1	0,62	0,56	0,59	**1**	0,64	0,39	0,49
Accuracy	0,61			**Accuracy**	0,67		

Second Method - Random Forest – Training 80 %							
Balanced		**Q-value < 0,1**		**Unbalanced**		**Q-value < 0,1**	
	Precision	**Recall**	**F1-score**		**Precision**	**Recall**	**F1-score**
0	0,56	0,88	0,68	**0**	0,60	0,86	0,71
1	0,70	0,30	0,42	**1**	0,39	0,14	0,21
Accuracy	0,59			**Accuracy**	0,57		
Balanced		**Q-value < 1**		**Unbalanced**		**Q-value < 1**	
	Precision	Recall	F1-Score		Precision	Recall	F1-Score
0	0,50	0,58	0,54	**0**	0,50	0,58	0,54
1	0,49	0,41	0,44	**1**	0,49	0,41	0,44
Accuracy	0,50			**Accuracy**	0,50		

Second Method - Random Forest – Training 85 %							
Balanced		**Q-value < 0,1**		**Unbalanced**		**Q-value < 0,1**	
	Precision	**Recall**	**F1-score**		**Precision**	**Recall**	**F1-score**
0	0,55	0,75	0,63	**0**	0,60	0,73	0,65
1	0,59	0,37	0,46	**1**	0,40	0,27	0,32
Accuracy	0,56			**Accuracy**	0,54		
Balanced		**Q-value < 1**		**Unbalanced**		**Q-value < 1**	
	Precision	Recall	F1-Score		Precision	Recall	F1-Score
0	0,56	0,77	0,65	**0**	0,66	0,84	0,74
1	0,62	0,39	0,48	**1**	0,61	0,37	0,46
Accuracy	0,58			**Accuracy**	0,65		

Table 26. Results of the Random Forest for the three scenarios in the second method

Considering the results obtained, with the same number of features selected compared to the previous methodology, we can make the following observations in terms of performance. Regarding the category of interest 1 (the label given to patients with the mutated RAS gene family) we see overall improvement for the Training 75 % and Training 85 % scenarios while decreasing slightly for the Training 80 % scenario.

5.5.3.1. Comparison of the collected results

Now it is possible to analyze and compare the features selected in the two methods proposed above, such as Bootstrapping and Random Forest with Feature importance. In particular, the features that are present in all scenarios among the various cases were determined as they might turn out to be of interest.

It thus turns out that for the case with genes selected by MutSig2CV with **q-value < 0,1**:

1. Features in common in the Training 75 % scenario with unbalanced case selected by Bootstrapping and Random Forest with Feature Importance: 91 (70%).
2. Common features in the scenario Training 75 % with balanced case selected by Bootstrapping and Random Forest with Feature Importance: 67 (60 %).
3. Common features in the Training 80 % scenario with unbalanced case selected by Bootstrapping and Random Forest with Feature Importance: 69 (68 %).
4. Common features in the Training 80 % scenario with balanced case selected by Bootstrapping and Random Forest with Feature Importance: 66 (67 %).
5. Common features in the Training 85 % scenario with unbalanced case selected by Bootstrapping and Random Forest with Feature Importance: 117 (75 %)
6. Common features in the 85 % Training scenario with balanced case selected by Bootstrapping and Random Forest with Feature Importance: 82 (61 %)

Further delving into the features in common within the same scenario, we observe that:

1. Feature in common in the Training scenario 75 % with q-value < 0.1 between balanced and unbalanced: 55
2. Feature in common in Training scenario 80 % with q-value < 0.1 between balanced and unbalanced: 52
3. Feature in common in Training scenario 85 % with q-value < 0.1 between balanced and unbalanced: 72

Finally, among the last three analyses with q-value < 0.1, there are 9 features in common:

- BCL9_Missense_Mutation_noClust

- PTEN_Splice_Site_noClust
- TGIF1_Nonsense_Mutation_noClust
- CHEK2_Nonsense_Mutation_noClust
- ELF3_Frame_Shift_Ins_noClust
- ERBB2_Missense_Mutation_noClust
- ERBB2_Splice_Region_noClust
- TGIF1_Splice_Site_noClust
- CHEK2_Silent_noClust.

Further comparison can be made with the 15 features determined in section 5.5.2.1, i.e., the features that belong to the category of patients with the RAS mutated gene family in common in the three scenarios. We then first select the features in common in the three scenarios for the q-value <0.1 and balanced case. The following 10 features in common with then result:

- BCL9_Missense_Mutation_noClust
- PTEN_Splice_Site_noClust
- CHEK2_Nonsense_Mutation_noClust
- ELF3_Frame_Shift_Ins_noClust
- RNF43_Frame_Shift_Del_noClust
- ERBB2_Missense_Mutation_noClust
- TGIF1_Splice_Site_noClust
- ERBB2_Splice_Region_noClust
- TGIF1_Nonsense_Mutation_noClust
- CHEK2_Silent_noClust.

Comparing them with the common 15 features determined in section 5.5.2.1, there are 6 features in common:

- PTEN_Splice_Site_noClust
- ELF3_Frame_Shift_Ins_noClust
- ERBB2_Missense_Mutation_noClust
- TGIF1_Splice_Site_noClust
- ERBB2_Splice_Region_noClust
- TGIF1_Nonsense_Mutation_noClust.

As for the case with genes selected by MutSig2CV with **q-value < 1**, it results:

1. Feature in common in scenarios Training 75 % with unbalanced case selected by Bootstrapping and Random Forest with Feature Importance: 45 (19 %).
2. Common features in the Training 75 % scenario with balanced case selected by Bootstrapping and Random Forest with Feature Importance: 13 (9 %).
3. Common features in the Training 80 % scenario with unbalanced case selected by Bootstrapping and Random Forest with Feature Importance: 53 (23 %).
4. Common features in the Training 80 % scenario with balanced case selected by Bootstrapping and Random Forest with Feature Importance: 11 (9 %).
5. Feature in common in the Training 85 % scenario with unbalanced case selected by Bootstrapping and Random Forest with Feature Importance: 62 (22 %).
6. Common features in the 85 % Training scenario with balanced case selected by Bootstrapping and Random Forest with Feature Importance: 11 (8 %).

Further delving into the features in common within the same scenario, it is observed that:

1. Feature in common in the Training scenario 75 % with q-value < 0.1 between balanced and unbalanced: 9.

2. Feature in common in the Training 80 % scenario with q-value < 0.1 between balanced and unbalanced: 6.
3. Feature in common in the Training 85 % scenario with q-value < 0.1 between balanced and unbalanced: 4.

Finally, among the last three analyses, there appear to be no common features.

5.6. Overall discussion and interpretation of the results

At the end of these analyses, we can make evaluations and observations for what concerns the feature selection and the prediction phase.

To begin with, proposing scenarios analysis with several cases proved to be optimal because it allowed us to evaluate both the performance of the proposed methods and the selected features. With a higher number of results, it allows considerations to be made about which methodology is more in line with the proposed objectives.

Moreover, we evaluate the robustness of single Lasso Logistic Regression by ranking in descending order according to the number of times they are selected in the Bootstrapping model. Keep in mind, however, that comparing scenarios a feature by the number of the position it occupies in the ranking can be misleading since the total number of features in each scenario is different, but it is still useful to have an overall view of the overall feature ranking. Therefore, the order in which the features are ranked should be taken into greater consideration. In particular, the two features that are present in the first six positions in the three scenarios result PTEN_Splice_Site_noClust and ERBB2_Splice_Region_noClust. We notice that both PTEN and ERBB2 genes belong to the patient with the RAS gene family mutated and, also, by consulting Malacard [13] (an open source database that contains useful information on diseases and annotations including comparability of scores between

diseases and disease-gene association) these two genes are recognized as being associated with colorectal cancer.

Furthermore, by comparing the 15 features in common in the three downstream scenarios (in the case with q-value < 0,1 and balanced) Lasso Logistic regression with the feature selected by Bootstrapping, we extract 6 features.

The application of different techniques (such as Bootstrapping and Mean Decrease in Impurity) is useful to determine the features that are mostly preserved in the proposed methods and furthermore to determine the robustness of the model. In particular, the comparison of the features selected by the two different approaches highlighted how those in the case of q-value <0,1 and unbalanced are most preserved.

For what concern the prediction phase, we propose different Machine Learning methods in multiple scenarios to choose the best strategy in terms of performances. To do this, we evaluate not only overall accuracy, since we experience weaker performance on RAS-mutated cases only, but also other metrics such as precision, recall and f1-score. Analyzing the metrics just mentioned in identifying the category of patients with the mutated RAS gene family, we observe overall better performance in the three proposed methodologies for case with MutSig2CV gene selection with q-value < 0.1. Regarding the balanced and unbalanced case, the best performance is observed in the proposed methodologies especially for the balanced case. Going into detail in the three scenarios, the best performance is observed in: Training 75 % falls in the case with q-value < 0.1 and unbalanced, Training 80 % falls in the case with q-value <0.1 and balanced and Training 85 % falls in the balanced case, but most with q-value < 1 (4 out of 6 models proposed). Evaluating the performance that absolutely turn out to be the best are those of the Training 75 % scenario with the case of genes selected by MutSig2CV q-value<0.1 unbalanced applying the alternative method by using Ridge Logistic Regression.

Performance evaluation of the different proposed machine learning models suggests the selection of d MutSig2CV genes with q value <0.1. In addition, since the absolute best performance for the unbalanced case, we also propose this case. Moreover, given that the absolute best performance for the unbalanced case, we also propose this case. Also, because it falls into the case with a reasonable percentage of training and test data.

After the evaluation of the methods, the scenario (and subcase) with the best performance is evaluated using the Python SHAP library. Then, the 130 features selected by Ridge Logistic regression are put in order of importance by this algorithm. Specifically, as explained in section 4.3.5, the evaluation of the overall impact of a given feature is determined by summing for all classes the average SHAP values, where SHAP values are calculated as the difference between the prediction of a model that uses the feature and one that does not.

The graph shows the top 9 features defined as most important by the algorithm.

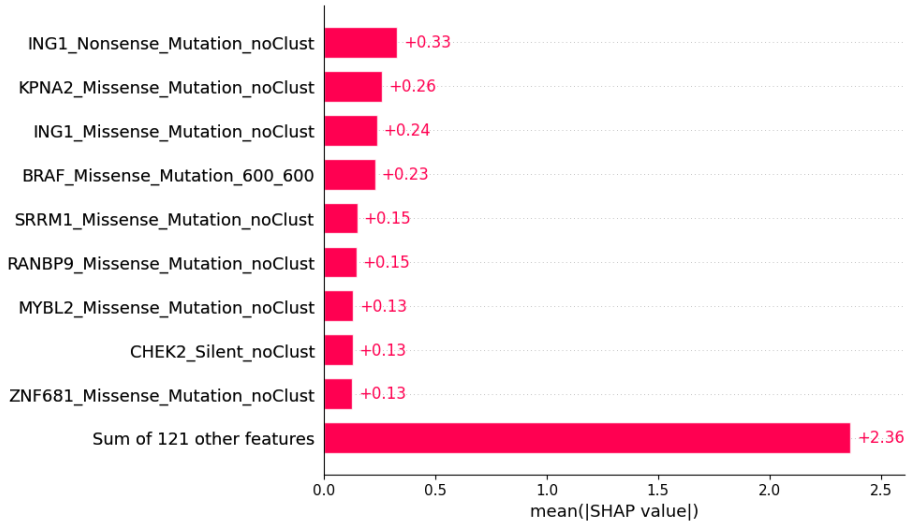


Figure 20. Results of SHAP algorithm

Analyzing the mutation type, it can be stated that Missense mutation is the most present (7/9), while Nonsense mutation (occupying the first place) is present in only 1/9 and also silent mutation in only 1/9.

Instead by analyzing the genes, genes from the dataset of patients with the mutated RAS family gene can be identified are ING1 and CHECK2, while those from the dataset of patients with the nonmutated RAS family gene are KPAN2, BRAF, SRRM1, RANBP9. Also by consulting Malacard, most of the genes listed above are recognized as being associated with colorectal cancer. The exceptions are SRRM1 and RANBP9 genes.

6 Conclusions

This thesis work is focused on colorectal cancer patients (CRC) with mutations in the RAS gene family (HRAS, KRAS, NRAS), since these patients do not respond to conventional therapies. The idea is to identify the most frequently co-occurring mutated genes, which could be potential targets for personalized therapy. To achieve this goal, we aim to assess and enhance a previous workflow which presented promising results. This thesis is focused on improving the encoding and selection phases, needed to transform the available mutational data into relevant features for Machine Learning techniques. In addition, it evaluates and enhances the previously proposed Data Science-based pipeline to better optimize the prediction phase and its results, together with the strategy required to identify relevant co-occurrent mutations.

Towards these aims, for what concerns the encoding phase we develop a method for identifying the category of patients, called hypermutants, to whom specific therapy is administered, would help in achieving the goal as these patients would thus be eliminated from the analysis. to achieve this goal, we first calculated the mutation rate (which indicates the rate at which a mutation occurs) using the Python library *pyensemble* which allows us to obtain the correct gene length on the number of bases in short computational times. then we developed a method for automatic threshold selection using another Python library called *kneelocator*.

Moreover, to determine genes and hotspots of interest, the investigation and a sensitive analysis for the significance thresholds of MutSig2CV and MutClustSW is done to improve the selection of genes and hotspots used to proceed with the application of Machine Learning models. To enhancing this step, we subdivide MutSig2CV results into subcases to identify a gene space of interest to proceed to

MutClustSW analysis. We decide to keep the sum of the genes that belong only to patients with the mutated RAS gene family and genes that belong only to patients with the unmutated RAS gene family because it thus makes it easy to identify which genes are associated with the patient category.

In addition, we create three scenarios of Supervised Learning containing 75, 80, and 85 percent of the patients for training phase, respectively, allowing us to make multiple simultaneous assessments and comparisons.

For what concerns the feature selection and prediction phases we evaluate several aspects. In the first place, we analyse the robustness of Lasso Logistic Regression models to more accurately select the most conserved features. Specifically, the features are ranked in descending order according to the number of times they are selected in a Bootstrapping setting including 100 Lasso Logistic Regression models. These analysis is performed considering for each configuration under exam three training/testing scenarios. Focusing on the features in common (in the case with q-value < 0,1 and balanced) we obtain 15 features: in particular, the two features that are present in the first six positions in all the three scenarios are PTEN_Splice_Site_noClust and ERBB2_Splice_Region_noClust, which belong to the patient with the RAS gene family mutated and are recognized by Malacard as being associated with colorectal cancer.

In addition, we apply Mean Decrease in Impurity as an alternative selection method based on feature importance to find other feature spaces and compare with those obtained by the Bootstrapping method. We observe that for all the three scenarios (in the case of q-value < 0,1 and unbalanced) the features selected are highly conserved (about 70 % of features in common). Furthermore, we extract the features in common between the Bootstrapping and Mean Decrease Impurity methods in the three scenarios in the balanced case and q-value <0.1 with a total of 10 features and we compare them with the previously mentioned 15 features in common in the three

scenarios (in the balanced case and q-value <0.1). This further comparison highlights the most robust and preserved features: PTEN_Splice_Site_noClust,ELF3_Frame_Shift_Ins_noClust,ERBB2_Missense_Mutation_noClust,TGIF1_Splice_Site_noClust,ERBB2_Splice_Region_noClust, TGIF1_Nonsense_Mutation_noClust.

Overall, using different methodologies in different scenarios optimized based on accuracy but evaluated also in terms of different performance metrics (including precision, recall, and f1-score) allows to eventually compare such strategies and extract the most interesting and shared features. We observe overall better performance in the three proposed methodologies for case with MutSig2CV gene selection with q-value < 0.1. Regarding the cases of balanced and unbalanced distribution of RAS-mutated and not-RAS mutated patients, better performances are reached in the proposed methodologies especially for the balanced case. The considerations just made about the performance of the models suggest further analysis of the 6 features that we found to be most robust and consistent (by comparing 10 features in common between the Bootstrapping and Mean Decrease Impurity methods in the three scenarios in the balanced case and q-value <0.1 and 15 features in common in the three scenarios in the case with q-value < 0,1 and balanced after applied Lasso Logistic Regression), thus offering a starting point for personalized therapies for CRC patients who do not respond to conventional therapies.

6.1. Future work

First, the analysis performed on patients with colorectal cancer can be carried out on another court of patients with other diseases with respective mutational data. Thus, the versatility of the proposed study turns out to be its great advantage.

We propose an analysis with a different gene space of interest from that proposed in our models. in particular, a gene space devoted entirely to patients with the mutated RAS gene family could lead to a more in-depth and centralized study of features in this category.

Moreover, since we experience weaker performance on RAS-mutated cases only, this may suggest that not the entire population of interest but just a sub-coohort is better recognized by specific features: future investigations could analyse the inherent heterogeneity of the mutated RAS subgroup and focus on one or more subgroups separately.

A further aspect that could be explored is to centralize the optimization on other evaluation metrics proposed rather than the accuracy, as this could lead to more focused considerations on the performance of the Machine Learning models, especially from the perspective given by the class of interest (e.g. precision/recall).

Finally, another curious consideration can be made by evaluating the performance and results obtained by considering multiple disease studies with their respective mutational data and seeing how different patients' selection might affect performance.

Bibliography

[1] Michele Bellomo, "Statistical and machine learning methods for discovering mutational signatures in RAS-mutated colorectal patients," 2022.

[2] A. Varabyou *et al.*, "CHESS 3: an improved, comprehensive catalog of human genes and transcripts based on large-scale expression data, phylogenetic analysis, and protein structure", doi: 10.1101/2022.12.21.521274.

[3] Breda Genetics, "Applications of Next Generation Sequencing," Apr. 16, 2016.

[4] "Tumore del colon e del retto."

[5] parenti e amici aimac - Associazione Italiana Malati di cancro, "Stadi e gradi del cancro del colon-retto," Mar. 14, 2014.

[6] M. S. Lawrence *et al.*, "Discovery and saturation analysis of cancer genes across 21 tumour types," *Nature*, vol. 505, no. 7484, pp. 495–501, 2014, doi: 10.1038/nature12912.

[7] M. S. Lawrence *et al.*, "Mutational heterogeneity in cancer and the search for new cancer-associated genes," *Nature*, vol. 499, no. 7457, pp. 214–218, 2013, doi: 10.1038/nature12213.

[8] J.-K. Rhee *et al.*, "Identification of Local Clusters of Mutation Hotspots in Cancer-Related Genes and Their Biological Relevance," *IEEE/ACM Trans Comput Biol Bioinform*, vol. 16, no. 5, pp. 1656–1662, Sep. 2019, doi: 10.1109/TCBB.2018.2813375.

[9] J. Gao *et al.*, "Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal," *Sci Signal*, vol. 6, no. 269, Apr. 2013, doi: 10.1126/scisignal.2004088.

[10] E. Cerami *et al.*, "The cBio Cancer Genomics Portal: An open platform for exploring multidimensional cancer genomics data," *Cancer Discov*, vol. 2, no. 5, pp. 401–404, May 2012, doi: 10.1158/2159-8290.CD-12-0095.

[11] M. Giannakis *et al.*, "Genomic Correlates of Immune-Cell Infiltrates in Colorectal Carcinoma," *Cell Rep*, vol. 15, no. 4, pp. 857–865, Apr. 2016, doi: 10.1016/j.celrep.2016.03.075.

[12] S. Seshagiri *et al.*, "Recurrent R-spondin fusions in colon cancer," *Nature*, vol. 488, no. 7413, pp. 660–664, Aug. 2012, doi: 10.1038/nature11282.

[13] N. Rappaport *et al.*, "MalaCards: An amalgamated human disease compendium with diverse clinical and genetic annotation and structured search," *Nucleic Acids Res*, vol. 45, no. D1, pp. D877–D887, Jan. 2017, doi: 10.1093/nar/gkw1012.

List of Figures

List if Tables